Unicenter

TCPaccess FTP Server Administrator Guide

Version 6.0

ca

Computer Associates

The Software That Manages eBusiness

Contents

Chapter 1: Architecture

Chapter 2: Customizing System Security

Chapter 3: Configuring FTP

Chapter 4: Using the System Management Facility (SMF)

Chapter 5: User Exits

Chapter 6: Troubleshooting the FTP Client and Server

Index

Chapter

1

Architecture

This chapter describes the system architecture of Unicenter TCPaccess FTP Server.

Unicenter TCPaccess FTP Server with the Unicenter NetMaster for File Transfer product enables you to transfer files to and from the mainframe using the File Transfer Protocol (FTP) over TCP/IP. Unlike conventional FTP servers, it includes a management interface that provides administrators with the ability to monitor and control FTP file transfers.

System Architecture

Unicenter TCPaccess FTP Server consists of the following components:

- An FTP client through which you can request file transfers

  The FTP client can run as a TSO command, a batch program, or under the OMVS shell.

- An FTP server to perform one side of the data transfer

- A second FTP server to perform the other side of the data transfer

  This can be the same server, although it is normally remote

- Components running in a Unicenter NetMaster for File Transfer region to monitor the transfer through events sent to it from the server

  The server communicating with the region must be in the same operating system image as the Unicenter TCPaccess FTP Server server.

- A Unicenter NetMaster for File Transfer data space that acts as a repository for FTP policy rules

Unicenter TCPaccess FTP Server System Architecture

(Any) FTP Server	FTP Server	**OPTIONAL**	Data Space

Rules

Events

FTP Client

NetMaster for File Transfer Region

Chapter 2
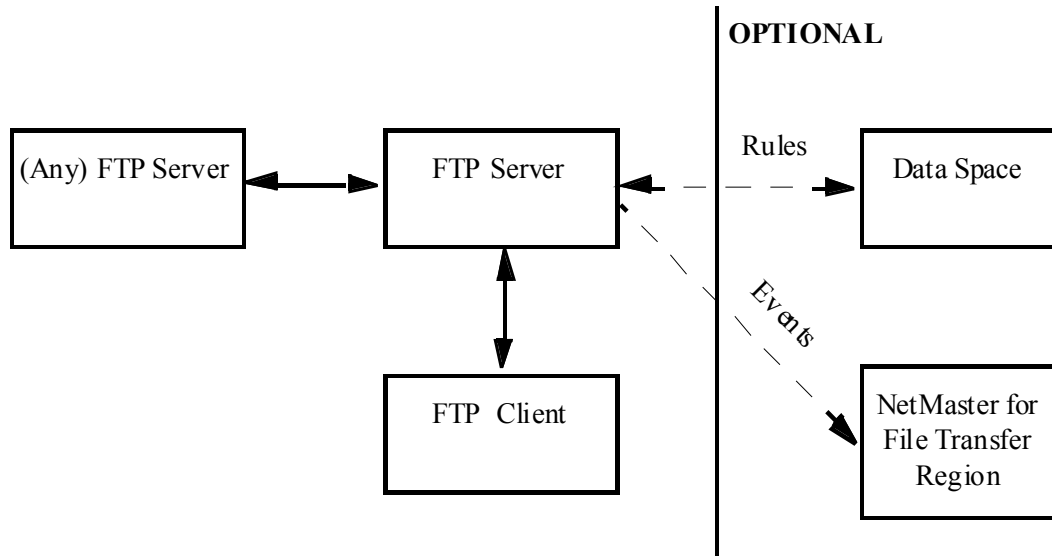
Customizing System Security

System security is an important consideration in data processing. Products like Access Control Facility 2 (eTrust CA-ACF2), eTrust CA-Top Secret, and Resource Access Control Facility (RACF) help many installations protect valuable data and preserve system integrity.

The following sections describe the security configuration procedures, as required by several security products: These preliminary tasks are described in the following sections:

■ CA-ACF2 Options

■ CA-Top Secret Options

■ RACF Options

Note: The examples use default class and profile names for illustration only; alternate name selection is possible. See the description of the SECURITY statement in the IJTCFG*xx* member for details.

In installations using external security systems, the security administrator usually establishes data access restrictions. The security administrator must ensure that Unicenter TCPaccess FTP Server does not circumvent these restrictions.

Unicenter TCPaccess FTP Server interfaces to the security system, via the SAF router, to perform these functions:

■ User ID and password validation

   The user ID and password are validated when sent to the FTP server. Validation occurs after connecting to FTP.

■ User privileges verification

   FTP uses the validated user security authority to determine if the user is permitted access to specific data sets. Access to data sets is determined by the security information associated with the user, not the security information for the job.

At eTrust CA-ACF2, RACF, and eTrust CA-Top Secret sites, the user ID associated with the FTP server job needs no special privileges assigned, such as NON-CNCL, OPERATIONS, or DASDVOL authority, or PPTNOPAS specified in the Program Properties Table. Also, the FTP server user ID does not need access to user data sets for FTP to function properly.

eTrust CA-ACF2 Options

This section describes the eTrust CA-ACF2 security options. See the appropriate eTrust CA-ACF2 procedures in the following sections, according to the version you are running.

- Customizing CA-ACF2 Version 5.2 or Earlier

- Customizing eTrust CA-ACF2 Version 6.0 or Later

Types of eTrust CA-ACF2 Security

Unicenter TCPaccess FTP Server uses these types of security with eTrust CA-ACF2:

- Signon security

    All user ID and password combinations are validated by eTrust CA-ACF2.

- Dataset security

    All FTP file transfers are validated by eTrust CA-ACF2.

- Source level security for FTP IP addresses

Because eTrust CA-ACF2 denies all access until permitted, additional steps are required to bring up Unicenter TCPaccess FTP Server at a site where eTrust CA-ACF2 is installed.

Customizing eTrust CA-ACF2 Version 5.2 or Earlier

To customize eTrust CA-ACF2 Version 5.2 or earlier, do the following:

1. Create a logon ID (LID) record to associate with the Unicenter TCPaccess FTP Server address space.

    Follow the installation procedures of your site to create an LID record; make sure the following parameters are set in the Privileges Section - Group 2:

    ```
    MUSASS
    NONO-SAF
    NO-INH
    BDT
    ```

    See the *eTrust CA-ACF2 Administrator Guide* for help on creating the LID record.

    Place the LID in the USER field of the startup JCL job card.

    You may have to set the NON-CNCL attribute in the LID record to be associated with the FTP server job. If your site runs *eTrust CA-ACF2* Version 5.2 with APAR TW95626, it is not necessary to place NON-CNCL in the FTP server LID record. If your site runs *eTrust CA-ACF2* version 5.2 without APAR TW95626, then you need to place NON-CNCL in the LID record.

2. Update the GSO records to allow SAF processing.

    Use the following commands to check the eTrust CA-ACF2 GSO record to see if SAF processing is allowed:

    ```
    ACF
    SET CONTROL(GSO)
    LIST OPTS
    ```

    If the output does not indicate that SAF is allowed, then enter these commands:

    ```
    CHANGE OPTS SAF REP
    END
    ```

3. Update the SAFMAPS records to let Unicenter TCPaccess FTP Server use SAF.

    The Unicenter TCPaccess FTP Server command class is AC#CMD. Unicenter TCPaccess FTP Server uses the SAF interface for general resource class checking. Use the following commands to check the SAFMAPS GSO records in the eTrust CA-ACF2 GSO record to see if SAF processing is allowed:

    ```
    ACF
    SET CONTROL(GSO)
    LIST SAFMAPS
    ```

    If the output does not indicate that SAF processing is allowed (SAF/-), then enter these commands:

    ```
    INSERT MAPS(SAF/AC#CMD)
    END
    ```

4. Follow these guidelines to set up the SAFPROT records to intercept Unicenter TCPaccess FTP Server calls to SAF:

    –   Enter the SAFPROT records exactly as shown.

    –   Do not change the SUBSYS(NETWRKIT) in the first SAFPROT record.

    –   NETWRK*IT* in the SUBSYS parameter relates to parameters on the SAF security calls, not the eTrust CA-ACF2 LID chosen by the site.

    To ensure that SAF calls from Unicenter TCPaccess FTP Server are processed by eTrust CA-ACF2, use the following commands to update the SAFPROT records in the GSO record:

```
ACF
SET CONTROL(GSO)
INSERT SAFPROT.AC#CMD CLASS(-) REP  CNTLPTS(ACSECPC) SUBSYS(NETWRKIT)
INSERT SAFPROT.FTP    CLASS(-) REP  CNTLPTS(FTP)     SUBSYS(-)
INSERT SAFPROT.FTP3   CLASS(-) REP  CNTLPTS(FTP3)    SUBSYS(-)
END
```

Customizing eTrust CA-ACF2 Version 6.0 or Later

To customize eTrust CA-ACF2 Version 6.0 or later, do the following:

1. Create a logon ID (LID) record to associate with the startup JCL.

    Follow the installation procedures of your site to create an LID record; make sure these parameters are set in the Privileges Section - Group 2:

```
MUSASS
NO-INH
BDT
```

    See the *eTrust CA-ACF2 Administrator Guide* for instructions on creating LIDs.

    Place the LID in the USER field of the startup JCL job card.

    If your site runs *eTrust CA-ACF2* Version 6.0 or higher, it is not necessary to set NON-CNCL in the Unicenter TCPaccess FTP Server LID record.

    Note: To prevent unauthorized users from attempting to use the production user ID for Unicenter TCPaccess FTP Server, the TCP base product rejects all attempts to log on to Unicenter TCPaccess FTP Server from programs like FTP and ACTEST that use this ID.

2. Update GSO records.

   Enter all commands exactly as shown. Do not change the
   SUBSYS=NETWRK*IT* in the first SAFDEF record. NETWRK*IT* in the
   SUBSYS parameter relates to parameters on the SAF security calls (not the
   LID chosen by the site).

   ```
   ACF
   SET CONTROL(GSO)
   INSERT CLASMAP.AC#CMD RESOURCE(AC#CMD) RSRCTYPE(SAF) ENTITYLN(8)
   CHANGE INFODIR TYPES(D-RSAF)
   INSERT SAFDEF.ACSECPC1 ID(ACSECPC) MODE(GLOBAL)  REP
          PROGRAM(BYPASS#1) RACROUTE(SUBSYS=NETWRKIT,REQSTOR=ACSECPC)
   INSERT SAFDEF.ACSECPC2 ID(ACSECPC) MODE(GLOBAL)  REP
          PROGRAM(BYPASS#2) RACROUTE(SUBSYS=NETWRKIT,REQSTOR=ACSECPC)
   INSERT SAFDEF.ACSECPC3 ID(ACSECPC) MODE(GLOBAL)  REP
          PROGRAM(BYPASS#3) RACROUTE(SUBSYS=NETWRKIT,REQSTOR=ACSECPC)
   INSERT SAFDEF.ACSECPC4 ID(ACSECPC) MODE(GLOBAL)  REP
          PROGRAM(BYPASS#4) RACROUTE(SUBSYS=NETWRKIT,REQSTOR=ACSECPC)
   INSERT SAFDEF.T051C ID(T051C) MODE(GLOBAL)   REP
          PROGRAM(T051C) RACROUTE(REQUEST=EXTRACT)
   INSERT SAFDEF.FTP ID(FTP) MODE(GLOBAL)
          PROGRAM(FPT) RACROUTE(REQUEST=EXTRACT)
   INSERT SAFDEF.FTP3 ID(FTP3) MODE(GLOBAL)
          PROGRAM(FPT3) RACROUTE(REQUEST=EXTRACT)
   ```

3. Use this command to build the INFODIR SAF records for Unicenter
   TCPaccess FTP Server:

   ```
   F ACF2,REBUILD(SAF),CLASS(R)
   ```

4. To activate the changes in steps 2 and 3, perform an IPL or issue a GSO
   console operator refresh. Use this command for the refresh:

   ```
   F ACF2,REFRESH(ALL)
   ```

eTrust CA-Top Secret Options

This section describes the types of security options available to sites running eTrust CA-Top Secret.

Types of eTrust CA-Top Secret Security

Unicenter TCPaccess FTP Server uses these types of security with CA-Top Secret:

- Signon security

  All user ID/password combinations are validated by eTrust CA-Top Secret.

- Data set security

  All file transfers under FTP are validated by eTrust CA-Top Secret.

- Source level security for FTP IP addresses

eTrust CA-Top Secret Customization

The Unicenter TCPaccess FTP Server address space functions as a true facility to eTrust CA-Top Secret. Use this setup to enable Unicenter TCPaccess FTP Server with eTrust eTrust CA-Top Secret:

1. Set up a Unicenter TCPaccess FTP Server facility entry with eTrust CA-Top Secret options as shown in this example:

```
TSS MODIFY FAC(USERx=NAME=NETWRKIT)
TSS MODIFY FAC(NETWRKIT=PGM=BYP)
TSS MODIFY FAC(NETWRKIT=ACTIVE,NOABEND,NOASUBM,NOAUDIT,AUTHINIT,ID=c)
TSS MODIFY FAC(NETWRKIT=NOINSTDATA,KEY=8,LCFCMD,LOCKTIME=0,NOLUMSG,
               LOG(NONE))
TSS MODIFY FAC(NETWRKIT=MULTIUSER,NORNDPW,RES,SIGN(M))
TSS MODIFY FAC(NETWRKIT=SHRPRF,NOSTMSG,NOTSOC,WARNPW,NOXDEF)
```

In the above example, the Unicenter TCPaccess FTP Server facility is named NETWRKIT. You can use any name up to eight bytes in length. If another name is used, then it must be substituted in the setup examples.

USERx can be any user-defined resource type available at the installation and the x value can be any keyboard character.

For ID=c, c is equal to a single alphanumeric that represents the facility for reporting purposes (see *facility* under eTrust CA-Top Secret control options).

Note: If you are using eTrust CA-Top Secret Version 4.2 or below, include the additional operands NOPSEUDO, NOMRO, and TENV in the facility definition; these operands are not supported in Version 4.3 and above.

2. Create the Unicenter TCPaccess FTP Server ACID.

   Build the ACID for the Unicenter TCPaccess FTP Server address space with the TSS CREATE command. The following command creates ACID FTPUSER to run as a started task:

   ```
   TSS CREATE(FTPUSER) NAME('FTPUSER ACID') FAC(STC) TYPE(USER)
   PASS(NOPW) DEPT(dept_name) MASTFAC(NETWRKIT)
   ```

3. Give ACIDs access to the Unicenter TCPaccess FTP Server facility.

   To permit ACID FTPUSER access to the Unicenter TCPaccess FTP Server facility (NETWRKIT), the security administrator must issue this command:

   ```
   TSS ADD(FTPUSER) FAC(NETWRKIT)
   ```

4. Give the user ID for the started task access to OMVS.

   ```
   TSS ADD(FTPUSER) UID(0) GROUP(groupid)
   ```

5. Give the user ID for the started task access to the FTP server HLQ.

   ```
   TSS ADD(FTPUSER) DSN(dsnpref)
   ```

6. The Unicenter TCPaccess FTP Server ACID must have authority to access the data sets it needs to function in the customer's environment.

   If you unloaded all the Unicenter TCPaccess FTP Server data sets with the HLQ *dsnpref* and the Unicenter TCPaccess FTP Server ACID is FTPUSER, then the security administrator can grant access to the Unicenter TCPaccess FTP Server ACID FTPUSER by issuing this command:

   ```
   TSS PERMIT(FTPUSER) DSN(dsnpref) ACCESS(UPDATE)
   ```

7. Set up the Unicenter TCPaccess FTP Server server as a started task.

   If the Unicenter TCPaccess FTP Server runs as a started task, then the relationship also must be established in the eTrust CA-Top Secret STC record. The following TSS ADDTO command connects the started task with the ACID defined by the TSS CREATE command. This example assumes that the Unicenter TCPaccess FTP Server PROC name is T051RUN and the ACID defined for use by the Unicenter TCPaccess FTP Server task is FTPUSER:

   ```
   TSS ADDTO(STC) PROC(T051RUN) ACID(FTPUSER)
   ```

8. Set up Unicenter TCPaccess FTP Server as a batch job.

   If the FTP server is run as a batch job, then the relationship is established by the USER= value coded on the job card. In this example, the FTP server job must be coded with USER=FTPUSER.

RACF Options

If a computer site runs RACF, the Unicenter TCPaccess FTP Server RACF interface automatically becomes active upon installation. This section describes the types of security options available to sites running RACF and how to customize security for Unicenter TCPaccess FTP Server.

Types of RACF Security

With RACF, these types of security are active in Unicenter TCPaccess FTP Server:

- Signon security

    All user ID/password combinations are validated by RACF

- Data set security

    All file transfers under FTP are validated by RACF

- Source level security for FTP IP addresses

Chapter

3

Configuring FTP

This chapter provides guidelines to help you customize Unicenter TCPaccess FTP Server for your site.

This chapter contains the following topics:

- Configuring the Unicenter TCPaccess FTP Server Address Space (T051CFIx)
- The File Transfer Protocol (FTP) SERVICE Statement (T051CFAx)
- FTP Configuration Parameters - FTP Statement
- Generic Attributes Table (GAT)
- FTP Support for SMF Activity Reporting and User Accounting
- FTPRSRCE and FTPLOGIN Exits
- Client FTP Configuration Data Sets

Configuring the Unicenter TCPaccess FTP Server Address Space (T051CFIx)

To start Unicenter TCPaccess FTP Server, you first need to set your software key in the T051CFI*x* member.

The IFSPARM Statement

The IFSPARM Statement in T051CFI*x* allows you to set GTF ID, timezone, and timing services.

```
IFSPARM      GTFID ( value )
             [ APPLICATIONS  (FTP) ]
             [ ARMELEMENT ( name ) ]
             [ PROMPT | NOPROMPT ]
             [ TIMER ( interval ) ]
             [ TIMEZONE ( ATLANTIC | EASTERN | CENTRAL |
                          MOUNTAIN | PACIFIC | char hours ) ]
             [ TRACENAME ( name ) ]
             [ VSREPORT ( interval ) | NOVSREPORT ]
```

GTFID (*value*)

Specifies the identification value for generalized trace facility (GTF) records produced by Unicenter TCPaccess FTP Server. This can be specified by a decimal value or a hexadecimal string.

Range: X'00' to X'4F' (0 to 79).

Default: X'4F' (79).

APPLICATIONS (FTP)

Specifies the applications to be available to the address space.

To run only the FTP Server, you must code APPLICATIONS(FTP). If you let the keyword value default, other application programs will be loaded as well, but will remain unused. Also, if you code PRFX= in your JCL parameters, you must specify a value of T01. Any other value will cause the FTP Server initialization to fail.

Note: The PRFX= JCL parameter is deprecated, and may not be available in a subsequent release of the product. It is suggested that you omit this parameter from your JCL, and use IFSPARM APPLICATIONS to specify which applications you want to run.

Alias: APPS

Default: STACK,TELNET,FTP

ARMELEMENT (*name*) Specifies the ARM element name to be assigned to the address space. It may contain the characters A-Z, 0-9, $, #, @, and _ (underscore). It may not begin with a number, the characters A-I, or the string SYS.

The special value "NONE" indicates that the address space is not to make itself eligible for automatic restart by ARM.

Default: NONE.

PROMPT | NOPROMPT

Specifies whether to prompt to issue message T00IJ13R, which requires a reply, before shutting down the address space.

Default: PROMPT.

TIMER (*interval*) Specifies the time interval for a timer interrupt. This specifies the resolution for timers used in this address space.

interval is in 0.01 seconds.

Range: 1 to 99.

Default: 10.

TIMEZONE (ATLANTIC | EASTERN | CENTRAL | MOUNTAIN | PACIFIC | *char hours*)

Specifies the time zone of local time or a character (that you can specify) and the number of hours before GMT (Greenwich Mean Time).

Default: EASTERN.

TRACENAME (*1*) Specifies the subsystem name of the component trace subsystem.

Default: ACTR.

VSREPORT (*interval*) | NOVSREPORT

Specifies whether to run the virtual storage report program. This program produces a message about virtual storage usage after the specified time to the T01LOG DD statement.

interval specifies how often, in minutes, this message is to be sent to the account log.

Default: five minutes, with a maximum of 1439 minutes and a minimum of one minute.

Default: NOVSREPORT.

Usage Notes

Timing Services

The IFS timing service, ITIME, allows interval timing to occur in modes other than primary task mode. The heart of the timing service is a fixed interval timer DIE that schedules SRBs in response to timers expiring.

Each timer is capable of tracking eight separate intervals. When the interval time expires, either an exit can be driven or an ECB posted.

The time interval (TIMER) for the DIE is configurable and has a range of 0.1 to 1 second. The time interval is specified in T051CFI*x*.

Note: Specifying a large interval decreases timer resolution. Specifying a very small interval increases timer overhead

Setting the Time Zone

To set the time zone for your system, specify your choice on the IFSPARM statement with the TIMEZONE parameter.

For example, to set the time zone to Eastern Standard Time:

```
IFSPARM TIMEZONE (EASTERN)
```

Setting the GTF ID

To set the Generalized Trace Facility (GTF) identification value for Unicenter TCPaccess FTP Server, specify the value on the IFSPARM GTFID statement.

 For example, to set the value as 17:

```
IFSPARM GTFID 17
```

Controlling Message Logging

The LOGGING statement in T051CFI*x* specifies the filtering for both print and console messages. Messages are filtered by component and level within component. For instance, specifying:

```
LOGGING PRINT((TC,RFEWISDT)) WTO((TC,FE))
```

prints all TCP messages, but only issues operator messages for fatal and error type TCP messages. Messages for other components are issued at their default levels for both print and operator messages.

The LOGGING statement also controls the spin attributes of the SYSOUT data set and the console routing codes of messages issued by Unicenter TCPaccess FTP Server.

LOGGING Statement

LOGGING   [CLASS (*class*)]
[DEST (*destination*)]
[FORMS (*name*)]
[PRINT ((*subparameter*) [, (*subparameter*) [, ...]])]
[ROUTCDE (*list*)]
[SPIN (LINES (*lines*) | MINUTES (*minutes*) | SYNC) |
NOSPIN]
[WRITER (*name*)]
[WTO ((*subparameter*) [, (*subparameter)* [, ...]])]

CLASS (*class*)   Specifies the SYSOUT class.

Default: The class specified as SOUT= *keyword* of PARM field

DEST (*destination*)   Specifies the SYSOUT destination.

Default: None.

FORMS (*name*)   Specifies the print form (corresponds to FORMS= on the T01LOG DD statement).

Default: None.

PRINT *(subparameter)*—Subparameters are processed left to right. Valid values are:

ALL	PRINT all messages, all types.
NONE	PRINT no messages.
ALL,types	PRINT given types for all components.
component,ALL	PRINT all messages for given component.
component,NONE	PRINT no messages for given component.
component,types	PRINT given messages for given component.

See Component and Message Type Definitions, for component and type specifications and defaults.

ROUTCDE *(list)*   Specifies the OS/390 routing codes for console messages.

list can be one or more valid OS/390 routing codes, separated by commas. Routing code ranges can be specified by separating them with a colon.

For example:

```
IFSPARM LOGGING ROUTECD(2)
IFSPARM LOGGING ROUTECD(3,4,8:11)
IFSPARM LOGGING ROUTECD(9:11)
```

Default: No routing code.

This means console messages are routed according to the defaults specified in the OS/390 SYSGEN.

Range: 1 to 16.

SPIN (LINES (*lines*) | MINUTES (*minutes*) | SYNC) | NOSPIN

Determines when the log file will be closed and reopened.

LINES	The number of lines in the log to trigger the spin; the maximum number of lines that can be specified is 2,000,000,000.
MINUTES	Number of minutes; the maximum number of minutes that can be specified is 357,913. **Note:** TIME is an alias for MINUTES.
SYNC	Specifying SYNC with a MINUTES parameter causes a SPIN on the next occurrence of the interval synchronized to the previous midnight. Specifying SYNC without a MINUTES parameter causes a SPIN on the hour, every hour.

Default: NOSPIN.

WRITER (*name*)   Specifies the member name of an installation-written program in the system library that the external writer loads to write the output data set. (Corresponds to WRITER= on the T01LOG DD statement.)

Note: Do not code INTRDR or STDWTR (or, for JES3, NJEWTR) as the writer name. These names are reserved for JES.

WTO ((*subparameter*) [, (*subparameter*) [, ...]])

Subparameters are processed left to right.

Valid values are:

ALL	WTO all messages, all types.
NONE	WTO no messages.
ALL,types	WTO given types for all components.
component,ALL	WTO all messages for given component.
component,NONE	WTO no messages for given component.
component,types	WTO given messages for given component.

The double parentheses surrounding both subparameter specifications are required for TSO parsing.

See Component and Message Type Definitions, for component and type specifications and defaults.

Component and Message Type Definitions

The following table displays the WTO and PRINT components and their message type defaults, according to message destination.

Message type is listed in the table and is specified as:

R	Response
F	Fatal
E	Error
W	Warning
I	Informational
S	Statistics
D	Debugging
T	Trace

Component Names and Message Defaults

Message	Description	Print Default Message Level	Console Default Message Level
AP	API support	RFEWI	RF
CF	Configuration	RFEWI	RFEW
CO	Commutator	RFEWI	RFE
EX	Exit interface	RFEWI	RFEWI
IF	IFS	RFEWI	RFEWI
IJ	IJT job step task	RFEWI	RFEWI
SF	Server FTP application	RFEWI	RF
SO	Socket API layer	RFEWI	RFEWI

Examples of Logging

These are some examples of syntax to log messages:

- To send everything to Write to Operator:

  `WTO(ALL)`

- To send the following messages to Write to Operator (WTO):

  - All messages for the IF component

  - Type Debug, Error, Warning, and Informational messages for the IJ component

    `WTO(NONE,(IF,ALL),(IJ,DEWI))`

- To print all type Debug and Error messages, but not anything for the SO component:

  `PRINT((ALL,DE),(SO,NONE))`

- To print nothing:

  `PRINT(NONE)`

Note: Parameters are processed left to right; for example:

PRINT(ALL,(IJ,I),NONE)

The final NONE subparameter overrides the ones before it.

Grouped Keywords

Here are some examples of the use of grouped keywords.

Print all messages; leave WTO, SYSOUT, and spin unchanged:

`LOGGING PRINT( ALL )`

WTO all messages; leave PRINT, SYSOUT, and spin unchanged:

`LOGGING WTO( ALL )`

Reset SYSOUT class to X; defaults DEST, FORM, and WRITER, and leave remaining keywords unchanged:

`LOGGING CLASS( X )`

Spin the log now; leave all specifications unchanged:

`LOGGING SPIN NOW`

Reset spin to every 5000 lines or 60 minutes, whichever comes first; leave remaining keywords unchanged:

`LOGGING SPIN( LINES( 5000 ) MINUTES( 60 ) )`

Print and WTO all messages, reset SYSOUT writer to TOLSTOY and defaults CLASS, DEST, and FORM. Also reset spin to every hour on the hour, ignoring line count, and spin the log now:

```
LOGGING PRINT( ALL ) WTO( ALL ) WRITER( TOLSTOY ) SPIN( SYNC ) NOW
```

Authorization Key

You must set your customer number and component authorization key correctly or Unicenter TCPaccess FTP Server will not operate. If you have not received these numbers, contact Technical Support.

Set the key in the AUTH statement in T051CFI*x*.

```
AUTH CUSTNUM ( cust_num )
KEY ( auth_key
```

CUSTNUM (*cust_num*)

Specifies your Unicenter TCPaccess FTP Server customer number for component authorization use.

KEY (*auth_key*)   Specifies the component authorization key. This key must be obtained from Technical Support.

Note: The key is case-insensitive and may be entered with spaces to improve readability.

Setting SMF Parameters

The TCP/IP stack generates SMF records at various processing points, such as connection establishment, connection termination, connection rejection, and failed connection attempts.

The INTERVAL parameter of the SMF statement determines the frequency (in minutes) with which periodic SMF statistics records, such as virtual storage utilization and driver statistics, are written.

The available record subtypes are described in the chapter "Using the System Management Facility (SMF)."

You can set your SMF parameters by using the SMF statement in T051CFI*x*.

```
SMF [ TYPE ( number ) ]
    [ SUBTYPE ( subtypes ) | ALL | NONE ]
```

TYPE (*number*) Specifies the SMF record type to be assigned to records generated by Unicenter TCPaccess FTP Server.

Any number from 128 to 255 not in use by your installation is available.

Default: 130.

SUBTYPE (*subtypes*) | ALL | NONE—Specifies record subtypes.

subtypes is the list of subtype numbers. The available record subtypes are described in the chapter "Using the System Management Facility (SMF)." Use a colon (:) to specify a range of subtypes.

ALL | NONE specifies if SMF recording is performed.

ALL   Specifies that all Unicenter TCPaccess FTP Server SMF record subtypes are generated.

NONE   Specifies that no SMF recording is performed.

For example:
```
SMF TYPE(135) SUBTYPE(20:22)
```

This SMF statement causes SMF record type 135 to be used, and subtypes 20 to 22 to be written.

Default: NONE.

SMF Usage Notes

Record Subtypes—The absence of an SMF statement in the configuration indicates that no SMF recording is performed.

Examples

These examples show the usage of the SMF statement:

```
SMF TYPE(189) ALL
SMF TYPE(241) SUBTYPE(21)
SMF NONE
```

Setting Exit Points

Use the EXIT statement to define global user exits to Unicenter TCPaccess FTP Server. Each EXIT statement defines an exit program to Unicenter TCPaccess FTP Server, and specifies at which exit point(s) the program will be invoked.

*Caution! The EXIT statement supplied in T051CFIx, containing exit points MFTPOLCY, MFTSTART, MFTPROGR, and MFTEND, is required for Unicenter TCPaccess FTP Server to function correctly and **must not** be modified.*

In addition to the exit points listed below, each exit program receives control at the INIT and TERM exit points. For more information on the usage of the Exit facility, see the "User Exits" chapter.

```
EXIT            PROGRAM ( program_name )
                [ PARM ( string ) ,]
                [ FTPCMND | NOFTPCMND | AUTOFTPCMND, ]
                [ FTPLOGIN | NOFTPLOGIN | AUTOFTPLOGIN, ]
                [ FTPRSRCE | NOFTPRSRCE | AUTOFTPRSRCE, ]
                [ LOG | NOLOG | AUTOLOG, ]
                [ SMF | NOSMF | AUTOSMF, ]
```

PROGRAM (*program_name*)

Identifies the name of the exit program to be invoked.

Note: This program must be made available to Unicenter TCPaccess FTP Server at startup.

PARM (*string*)   Any string of data that is desired. This string will be passed, uninterpreted, to the program at the INIT exit point.

FTPCMND | NOFTPCMND | AUTOFTPCMND

>Specifies whether the FTPCMND exit point will be called (FTPCMND) or not (NOFTPCMND), or whether it will be left to the INIT exit to specify (AUTOFTPCMND).

>Default: AUTOFTPCMND.

FTPLOGIN | NOFTPLOGIN | AUTOFTPLOGIN

>Specifies whether the FTPLOGIN exit point will be called (FTPLOGIN) or not (NOFTPLOGIN), or whether it will be left to the INIT exit to specify (AUTOFTPLOGIN).

>Default: AUTOFTPLOGIN.

FTPRSRCE | NOFTPRSRCE | AUTOFTPRSRCE

>Specifies whether the FTPRSRCE exit point will be called (FTPRSRCE) or not (NOFTPRSRCE), or whether it will be left to the INIT exit to specify (AUTOFTPRSRCE).

>Default: AUTOFTPRSRCE.

LOG | NOLOG | AUTOLOG

>Specifies whether the LOG exit point will be called (LOG) or not (NOLOG), or whether it will be left to the INIT exit to specify (AUTOLOG).

>Default: AUTOLOG.

SMF | NOSMF | AUTOSMF

>Specifies whether the SMFEXIT exit will be called (SMF) or not (NOSMF), or whether it will be left to the INIT exit to specify (AUTOSMF).

>**Note:** A sample SMFEXIT is provided in the SAMP data set.

>Default: AUTOSMF.

GLOBAL Parameters (T051CFAx)

The first statement of T051CFA*x* is the GLOBAL statement, which defines global options for the FTP server. It is followed by statements specific to the application. For more information on the statements that follow (FTP, GAT, and SERVICE), see the specific application chapters.

GLOBAL Statement

```
GLOBAL
[ MIGVOL ( volser ) ]
[ SMS | NOSMS ]
[ SVC99WTO | NOSVC99WTO ]
[ TRANTBL ( tablename ) ]
```

MIGVOL (*volser*)   Specifies the volume serial, such as HSM, used by your storage management software for migrated data sets.

Default: MIGRAT.

SMS | NOSMS   Specifies that SMS is to control allocation of new data sets. If SMS is specified, then many of the default DCB allocation parameters are not specified by Unicenter TCPaccess FTP Server. If NOSMS is coded, then Unicenter TCPaccess FTP Server sets the default for many of the DCB allocation parameters. If SMS is coded, be sure to verify that SMS will control allocation of data sets.

Default: NOSMS.

SVC99WTO | NOSVC99WTO

Specifies whether to ask OS/390 to produce additional diagnostic WTO messages when processing an SVC99 (dynamic allocation) request. This can be useful in diagnosing allocation problems.

Default: NOSVC99WTO.

TRANTBL (*tablename*)   Specifies the translation table load module to be used by the control connection for the service.

tablename can be one of the following:

ENGLISH
DANISH
FCANADA
FRENCH
GERMAN
GSWISS
ITALIAN
SPANISH
SWEDISH

Default: The translation table specified by TRANTBL parameter on the TELNET statement in the TCPCFG*xx* configuration member.

Note: If no parameter is specified, then ENGLISH is used.

Example

This example is typical for an installation with minimal changes from the sample. SVC99 debugging is enabled.

```
GLOBAL  MIGVOL(MIGRAT)
        SVC99WTO
        TRANTBL(ENGLISH)
```

The File Transfer Protocol (FTP) SERVICE Statement (T051CFAx)

The SERVICE statements define the user-level services provided by Unicenter TCPaccess FTP Server and the well-known port numbers. The most common ones are TELNET, VTAMAPPL, FTP, LPR, and USMTP/SSMTP.

```
SERVICE      NAME ( FTP )
             MODULE ( T051S )
             PORT ( number )
             [ QLISTEN ( number ) ]
```

NAME (FTP)

Specifies the FTP server.

Note: This is the only valid value.

MODULE (T051S)

Specifies the name of the primary load module (one- to eight-characters) to be started as the user-level protocol process (ULPP).

Note: T051S is only valid value.

PORT (*number*)

Specifies the well-known port number for the ULPP (1 to 4095).

Note: There is no default. It should correspond to the value specified in the FTP PORT parameter.

Port numbers below 1024 require the user ID associated with the server started task or batch job to have the authority level of O/E super user.

QLISTEN (*number*)

Specifies the number of entries to be allocated in the Queued Listen Table.

This value indicates the number of unspecified (wild) listens to be queued to determine whether the connection should be accepted or rejected.

Default: Five.

Example

```
SERVICE  NAME(FTP)
         MODULE(T051S)
         PORT(21)
```

This is for the standard server FTP (port 21).

FTP Configuration Parameters - FTP Statement

Use the FTP statement to define the configuration parameters for the File Transfer Protocol (FTP) server.

```
FTP        [ ABORT | NOABORT ]
           [ ACCOUNT ( userid account ) ]
           [ ACCTREQ | NOACCTREQ ]
           [ APPEND | NOAPPEND ]
           [ AUTORECALL | SITERECALL | NORECALL ]
           [ CD | NOCD ]
           [ CHARSET ( sbcs dbcs { CHAR | DBCS | MIX } ) ]
           [ CONDDISP ( CATLG | DELETE ) ]
           [ DATACLOSE ( time ) ]
           [ DATAIDLE ( number ) | IDLE ( number ) ]
           [ DATAOPEN ( time ) ]
           [ DATAPORTTOS ( tos_value ) ]
           [ DATASETMODE| DIRECTORYMODE ]
           [ DEBUG ]
           [ DEFGAT ( name ) ]
           [ DEFPRFX ( TSOPREFIX | NONE | USERID | userid ) ]
           [ EXPDTCHK | NOEXPDTCHK ]
           [ GUEST ( logonid password ) ]
           [ HSM ( number1 number2 ) ]
           [ IBUF ( numbuf bufsize max_numbuf max_bufsize ) ]
           [ ISPFENQ | NOISPFENQ ]
           [ ISPFRES | NOISPFRES ]
           [ JESPUTGETTO (time) ]
           [ JESFILTER (OWNERID | USERID) ]
           [ KEEPALIVETIMER ( number ) ]
           [ LISTFMT ( OLD | IBM | SHORT ) ]
           [ LKEDRES | NOLKEDRES ]
           [ MIGRATEVOL ( volser ) ]
           [ MOUNT ( def_wait_time, max_wait_time ) ]
           [ NDAB ( number1 number2 ) ]
           [ NLSTCASE ( UPPER | LOWER ) ]
           [ NOPERSIST | PERSIST ]
           [ OBUF ( numbuf bufsize max_numbuf max_bufsize ) ]
           [ OUTLIM ( number ) ]
           [ OVERWRITE | NOOVERWRITE ]
           [ PAD ( B | O | Z | C | hex1 [ B | O | Z | CC | hex2 ] ) ]
           [ PORT ( number ) ]
           [ RCVRID (name)
           [ RDW |NORDW ]
           [ RELEASE | RLSE | NORELEASE | NORLSE ]
           [ SITEALLOC ( NEW | ALL ) ]
           [ SITEREPLY ( 200 | 500 ) ]
           [ SITEOVERWRITE ]
           [ STRIP | NOSTRIP ]
           [ TABS ( number ) ]
           [ TRANTBL ( tranname ) ]
           [ UMASK (ooo) ]
           [ UNIT ( name ) ]
           [ WRAPRECORD | NOWRAPRECORD ]
           [ VOLUME ( volser ) ]
```

ABORT | NOABORT   Selects data connection termination when a permanent I/O (exception) error is detected during data transfer.

NOABORT   Causes a TCP FIN to be sent.

ABORT   Causes a TCP RESET.

Use caution when selecting ABORT because some remote hosts close the control connection as well as the data connection.

Default: NOABORT.

ACCOUNT (*userid account*)

Specifies the user ID and account number in sublist notation (one– to eight-characters each) to be used for FTP overhead accounting.

Default: (OVERHEAD 0000).

ACCTREQ | NOACCTREQ

Specifies whether FTP is to prompt for account data at login.

NOACCTREQ   FTP does not require account data at login.

ACCTREQ   FTP prompts for account data at login.

Default: NOACCTREQ.

APPEND | NOAPPEND

Specifies whether an APPEND operation to an existing OS/390 data set is allowed.

Default: APPEND.

AUTORECALL | SITERECALL | NORECALL

Specifies the HSM recall environment.

AUTORECALL   Migrated data sets are recalled automatically when referenced.

SITERECALL   User is required to issue the SITE RECALL command before a migrated data set can be recalled.

NORECALL   All requests for migrated data sets fail.

NORECALL and HSM() are mutually exclusive.

Default: AUTORECALL.

CD | NOCD   Specifies whether to allow directory commands.

Certain client FTP processes (for example, Novell LAN workplace) do not work properly with directory commands enabled.

Default: CD.

CHARSET (*sbcs dbcs* { CHAR | DBCS | MIX })

Specifies single and double byte character set, and default server translation mode for the data transfer port.

sbcs   Defines the single byte character set used for ASCII to EBCDIC and EBCDIC to ASCII single byte translation.

dbcs   Defines the double byte character set used for ASCII to EBCDIC and EBCDIC to ASCII double byte translation.

CHAR   Defines the default character translation mode as single byte.

DBCS   Defines the default character translation mode as double byte.

MIX   Defines the default character translation mode as single byte and/or double byte.

Note: Use the TRANTBL parameter for the control port.

Default: NGLISH single byte character set, with no double byte character set, and with CHAR mode.

CONDDISP (CATLG | DELETE)

> Specifies the default conditional disposition of new data sets when a STOR operation fails.
>
> Default: CATLG.

DATACLOSE (*time*)   Specifies the time, in seconds, the server will wait for a data connection to close.

> Minimum value: 30 seconds.
> Maximum value: 86,340 (1439 minutes).
>
> Default: 60.

DATAIDLE (*number*) | IDLE (*number*)

> Specifies the maximum amount of time (60 to 86340), in seconds, FTP should wait for transfer of one buffer. See the descriptions of the IBUF and OBUF operands in the section, Usage Notes for the FTP Statement for details.
>
> **Note:** This is used to detect a failed remote host during data transfer.
>
> Default: 60.

DATAOPEN (*time*)   Specifies the time, in seconds, the server will wait for a data connection to open. The minimum value is 30 seconds. The maximum value is 86,340 seconds (1439 minutes).

> Default: 1800.

DATAPORTTOS (*tos_value*)

> Specifies a value to be used as the TOS (type of service) in the IP header of FTP traffic on the data port.
>
> Range: 0 to 255.
>
> Aliases: DPTOS and TOS.
>
> Default: The *tos_value* for this parameter, unless the IP statement contains an overriding TOS default.

DATASETMODE | DIRECTORYMODE

> Specifies whether directory output (LIST/NLST) will be displayed in data set mode or directory mode.
>
> Default: DATASETMODE.

DEBUG   Activates the debugging option to provide tracing information on commands submitted during the FTP session.

                         Note: The DEBUG option can be used interchangeably with the TRACE and DEBUG commands.

DEFGAT (*name*)   Specifies a data set type in a GAT (Generic Attributes Table) statement to be used as a default attribute entry (see the section, Generic Attributes Table (GAT).

                         Default: None.

DEFPRFX (TSOPREFIX | NONE | USERID | *userid*)

                         Specifies installation default prefix (high level qualifier(s)) when data sets are not fully qualified.

                         TSOPREFIX   Specifies that the TSO prefix from the security system's database be used as the default, working directory. Use this option only if the security system supports TSO profile information (specifically, the data set name prefix) and supports the SAF interface to retrieve it.

                         NONE   Prefixing is not performed.

                         USERID   User's signon ID is used as a prefix.

                         userid   Installation defined character string.

                         Default: USERID.

EXPDTCHK | NOEXPDTCHK

                         Specifies whether FTP is to perform expiration date checking. NOEXPDTCHK specifies that no date checking is performed.

                         If you do not set up an automated response to IBM WTOR IEC507D, then you should code EXPDTCHK or let it default to that value. Otherwise, Unicenter TCPaccess FTP Server hangs until WTOR IEC507D receives a reply from an operator.

                         Default: EXPDTCHK.

GUEST (*logonid password*)

>>> Specifies the logon user ID and password (one- to eight- characters each) to be used for anonymous logons. This is passed to the security system in place of the default user ID, ANONYMOUS, and the default password, GUEST.

>>> Default: (GUEST *password*).

HSM (*number1 number2*)

>>> Specifies the default and maximum wait times for an HSM recall.

>>> *number1*   Defines the default wait time in minutes an HSM recall has to complete before the request is canceled. The integer value specified in the SITE RECALL command overrides the default.

>>> *number2*   Defines the maximum amount of time in minutes an HSM recall can wait before the request is canceled. This number is used to validate the integer specified in a SITE RECALL command. It must not exceed 1439.

>>> See the section, Usage Notes for the FTP Statement, for information about timeouts and FTP transfers to tape.

>>> Default: (5 1439).

IBUF (*numbuf bufsize max_numbuf max_bufsize*)

>>> Specifies the number of network input buffers (*numbuf*) and the buffer size (*bufsize*) to be used as defaults, and the maximum buffer number (*max_numbuf*) and buffer size (*max_bufsize*) that may be specified on the SITE command.

>>> **Note:** If the maximum values are not specified or are set to zero, the SITE IBUF command is not permitted.

>>> Each number must not exceed 32767.

>>> Default: (16 2048).

ISPFENQ | NOISPFENQ

Specifies that the ISPF enqueue facility be activated (ISPFENQ) or deactivated (NOISPFENQ).

See the section, Generic Attributes Table (GAT), for more parameters for using PDS enqueue.

Default: NOISPFENQ.

ISPFRES | NOISPFRES Enables (ISPFRES) or disables (NOISPFRES) the RESERVE logic for the SPFEDIT ENQ, if the volume on which the PDS resides is shared by multiple systems (UCB shared bit ON). This assures data integrity while the PDS you are accessing is being simultaneously accessed by an ISPF user from another system.

See the section, Generic Attributes Table (GAT), for more information to support PDS enqueue.

Default: NOISPFRES.

JESPUTGETTO (*time*)   Specifies time, in seconds (0 to 86400), the server will wait before a PUTGET (automatic retrieval of a submitted job) timeout occurs.

Default: 600 (10 minutes).

JESFILTER (OWNERID | USERID)

Specifies a filter to limit user access to batch jobs.

OWNERID   Only jobs that have an owner ID of the logon user are allowed.

USERID   Only jobs that have a job name of user ID plus one character are allowed.

Default: OWNERID.

KEEPALIVETIMER (*number*)

Specifies the time interval, in minutes, for TCP keepalive packets on the control connection when Unicenter TCPaccess is the TCP/IP provider.

Alias: KATIMER

Range: 0 (no keepalive) – 1439 (1439 minutes).

Default: Zero (do not use keepalive).

LABEL (BLP | NOBLP, NL | NONL)

Controls the types of label processing permitted.

BLP   Bypass label processing permitted.

NOBLP   Bypass label processing not permitted.d

NL   Non-label tapes permitted.

NONL   Non-label tapes not permitted.

Default: NOBLP, NONL.

LISTFMT (OLD | IBM | SHORT)

Specifies whether output from the data set LIST command will be in the old Network*IT* FTP format, in the IBM-standard format, or in a shortened IBM-compatible format. The short format leaves out data set extents and tracks allocated, but improves LIST response time.

Certain PC-based client FTP packages expect the LIST output from a host configured as OS/MVS to be in standard IBM format.

Default: OLD.

LKEDRES | NOLKEDRES

Enables (LKEDRES) or disables (NOLKEDRES) the RESERVE logic for the SYSIEWLP ENQ, if the volume on which the PDS resides is shared by multiple systems (UCB shared bit ON). This assures data integrity while the PDS you are accessing is being simultaneously accessed by the linkage editor from another system.

See the section, Generic Attributes Table (GAT), for more parameters for using PDS enqueue.

Default: NOLKEDRES.

MIGRATEVOL (*volser*) Specifies the default volume serial number for migrated data sets.

Default: MIGRAT.

MOUNT (*def_wait_time*, *max_wait_time*)

> Indicates tape mounts are supported if the *def_wait_time* parameter has a non-zero value. Omission of this parameter, or specifying MOUNT(0), turns off tape support.
>
> | | |
> |---|---|
> | *def_wait_time* | Indicates the default time in minutes that the FTP protocol allows for any operation involving operator intervention on a tape unit. |
> | *max_wait_time* | Indicates the highest value permitted on the SITE MOUNT() command. |
>
> When a tape drive unit is not available, FTP retries the request every 30 seconds until this time expires.
>
> See the section, Usage Notes for the FTP Statement, for more information about timeout considerations.
>
> Default: (0,1439).

NDAB (*number1 number2*)

> Specifies the number of DASD (DISK) buffers used by FTP for reading or writing disk data sets by default (*number1*), and the maximum buffer number permitted on the SITE NDAB command (*number2*).
>
> If *number2* is not specified or is set to zero, the SITE NDAB command is not permitted. The size of each buffer is the block size of the data set. At least two should be specified to overlap I/O. OS/390 has an upper limit of 99 buffers.
>
> Default: Four.

NLSTCASE (UPPER | LOWER)

> Specifies whether the output from an NLST command will be uppercase or lowercase. If LOWER is specified and the data set or member list is part of the current directory, the names are returned in lower case.
>
> NLSTCASE(LOWER) is supplied to facilitate MGET functions from FTP clients on systems that use lowercase file names.
>
> Default: UPPER.

NOPERSIST | PERSIST Specifies whether SITE parameters will be reset following data transfer.

If NOPERSIST is used, then all SITE parameters are reset after each data transfer.

If PERSIST is used, then all SITE parameters remain in effect until explicitly changed by subsequent SITE commands, or reset with SITE RESET.

Default: NOPERSIST.

OBUF (*numbuf bufsize max_numbuf max_bufsize*)

Specifies the number of network output buffers (*numbuf*) and the buffer size (*bufsize*) to be used as defaults, and the maximum buffer number (*max_numbuf*) and buffer size (*max_bufsize*) that can be specified on the SITE command.

Note: If the maximum values are not specified or are set to zero, the SITE OBUF command is not permitted.

Each number must not exceed 32767.

Default: (16 2048).

OUTLIM (*number*) Specifies the maximum number of records which can be submitted to the system internal reader in one file transfer (SITE SUBMIT).

This parameter is used for the OUTLIM parameter to JES.

Default: 250000.

OVERWRITE | NOOVERWRITE

Specifies whether a file transfer to an existing data set can write over that data set.

Default: OVERWRITE.

PAD (B | O | Z | C | *hex1* [B | O | Z | CC | *hex2*])

Pad character(s) to pad network records or lines to fixed-length logical records when data is stored (by STOR or APPE) or deleted from fixed- length logical records when data is retrieved (by RETR).

The first value is the sbcs pad character; the second value is the dbcs pad character.

B	Pad with blanks.
O	Pad with all ones.
Z	Pad with zeros.
C	A one-byte character other than B, O, or Z (sbcs).
CC	A two-byte character other than B, O, or Z (dbcs).
hex1	Any valid one-byte hexadecimal value (sbcs), represented by two hexadecimal digits.
hex2	Any valid two-byte hexadecimal value (dbcs), represented by four hexadecimal digits.

Default: (B Z).

PORT (*number*)
Specifies the port for the control connection. Normal usage is port 21.

Port numbers below 1024 require the user ID associated with the server started task or batch job to have the authority level of O/E super user.

Default: 21.

RCVRID (*name*)
Specifies the *receiver_id* (one- to eight-characters) used to uniquely identify the server to NetMaster for File Transfer. If you change the value of this parameter, then you must also change the Receiver ID in NetMaster for File Transfer.

Default: $RFFTEVR.

RDW/NORDW
Specifies whether Record Descriptor Words (RDWs) will be sent as data for RECFM=VB and RECFM=VBS files.

If RDW is selected, the RDW will be sent for binary, ASCII, or EBCDIC transfers.

Default: NORDW.

RELEASE | RLSE | NORELEASE | NORLSE

Specifies whether unused space in a data set being written to should be released.

Default: RELEASE.

SITEALLOC (NEW | ALL)

Specifies whether certain SITE command data set allocation parameters will be applied to all data sets or only to newly allocated data sets.

If NEW is specified, the SITE command parameters VOLUME, UNIT, LRECL, BLKSIZE, RECFM, and DCBDSN will be applied only to the newly created data sets.

Default: ALL.

SITEREPLY (200 | 500) Specifies whether a 200 or a 500 reply code will be used when errors occur in processing a SITE command.

Default: 500.

SITEOVERWRITE   SITEOVERWRITE specifies that files will not be overwritten unless a SITE OVERWRITE command is received.

STRIP | NOSTRIP   Specifies whether pad characters will be stripped from fixed-length logical records when data is retrieved (by RETR).

Default: STRIP.

TABS (*nn*)   Specifies the tab skip value, where *nn* is a digit between 0 and 25.

Default: Eight.

TRANTBL (*tranname*)   Specifies the translation table load module to be used by the control port. *tranname* can be one of the following:

ENGLISH
DANISH
FCANADA
FRENCH
GERMAN
GSWISS
ITALIAN
SPANISH
SWEDISH

Use the CHARSET parameter for the data port translation.

Default: the translation table specified by the TRANTBL parameter on the GLOBAL statement in the application configuration member.

UMASK (*ooo*)   Allows you to specify a three-character octal number for umask values. Sets the file access defaults for the session.

If this is not specified, it defaults to 022.

Note: If no SITE commands are issued and this is specified in the application configuration member, then this is the session default.

UNIT (*name*)   Specifies a generic unit name of default direct access volumes to be used for dynamic allocation.

Default: None.

WRAPRECORD | NOWRAPRECORD

Specifies whether, when storing, the server will wrap or truncate network records that exceed LRECL.

Default: WRAPRECORD.

VOLUME (*volser*)   Specifies the default volume serial number.

Default None.

Usage Notes for the FTP Statement

You may need to change these parameters on the FTP statement:

GUEST—If you want to allow anonymous/guest capability, enter a valid ID and password combination that is known to your security system.

ACCOUNT—If you are implementing FTP accounting, enter a user ID and account number to be used for overhead accounting.

UNIT—Enter a generic unit name to be used when FTP creates new data sets.

CHARSET—Enter the default FTP character translation table to be used.

ACCTREQ—This is required for FTP accounting, to prompt for the account number.

NOCD—If you use Novell LAN Workplace, you must disable the CD/CWD facility.

Customizing the FTP Greeting

You can customize the Unicenter TCPaccess FTP Server server greeting for your site. When you sign on to the Unicenter TCPaccess FTP Server server, this is the normal display response:

```
230 Logged in - User=uuuuuuuu  Working directory "dddddd"
230 The local host for the control connection is nnn.nnn.nnn.nnn.
```

The Unicenter TCPaccess FTP Server administrator can implement a custom login greeting by placing a member named FTPGREET in the HELP data set. Unicenter TCPaccess FTP Server searches for this member on the first login to the   FTP server after startup. If it is found, its contents are displayed after each successful login to Unicenter TCPaccess FTP Server.

This is how the greeting displays:

```
230--- GREETINGS ---
 text_of_FTPGREET_member
230 Logged in - User=uuuuuuuu  Working directory "dddddd"
230 The local host for the control connection is nnn.nnn.nnn.nnn.
```

Anonymous Logons

The Unicenter TCPaccess FTP Server server supports anonymous logon using the standard ANONYMOUS/GUEST user name and password. The logon ID and password provided by the GUEST operand is the user ID/password combination supplied to the security system for such logons. In other words, network users logging on to Server FTP using the ANONYMOUS/GUEST logon procedure assume the identity and privileges of the user ID specified in the GUEST operand. If anonymous logon is used, the user must be defined to your security system with specific access privileges.

Timing Out

FTP times out a data transfer request if the remote does not complete the data connection in a certain time. If the remote is another OS/390 system using tapes, or recalling a data set, it will require a tape mount on the remote system before it can complete the data connection. For this reason, FTP uses the *longer* of MOUNT or HSM times, or 30 minutes if neither tape support nor HSM support is configured.

The problem of a remote region using tape data sets should also be considered when configuring DATAIDLE time. If a remote region is reading a multivolume data set, for instance, it may have to stop the data transfer between volumes while the next tape is mounted. The DATAIDLE time may expire while this is happening.

Parameters for Previous Users of IBM TCP/IP for MVS FTP Server

For previous users of IBM TCP/IP for MVS FTP server, these configuration parameters must be specified for product transparency:
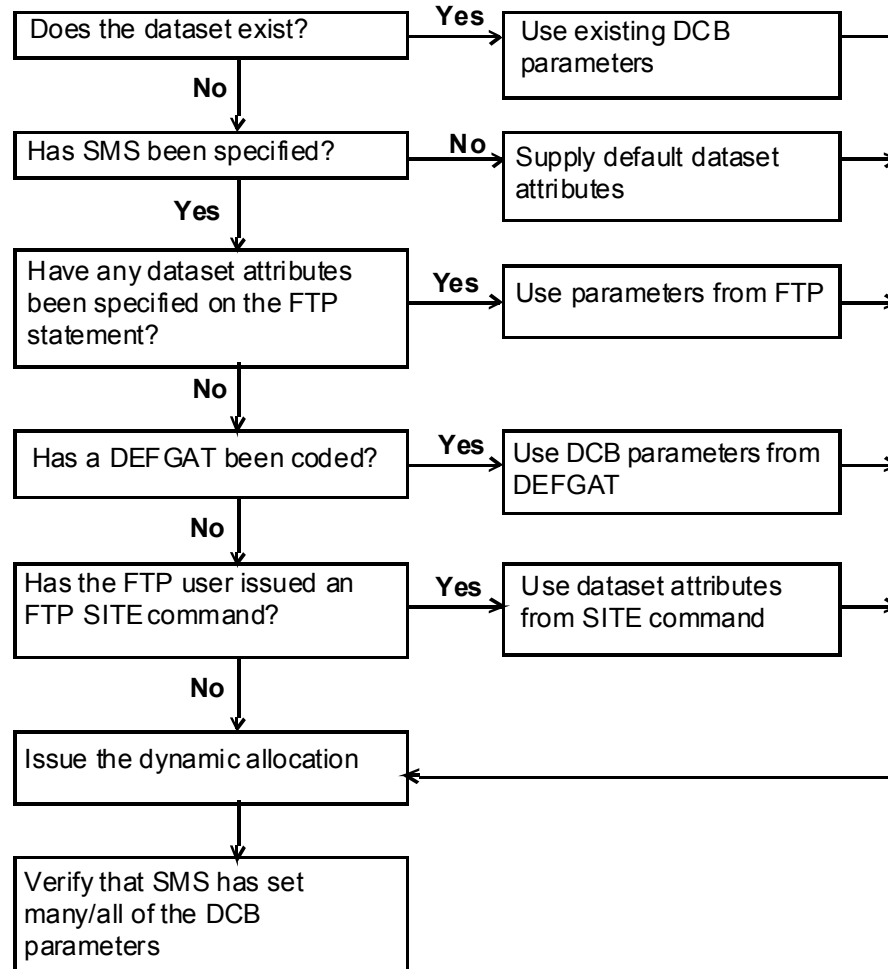
```
FTP SITEALLOC(NEW)
   DEFPRFX(TSOPREFIX)
     SITEREPLY(200)
     DATACLOSE(120)
     DATAIDLE(120)
     DATAOPEN(60)
     TABS(0)
     PERSIST
```

SMS Control by GLOBAL Statements

When the SMS parameter is specified on the GLOBAL statement of the APPCFG*xx* member, the Unicenter TCPaccess FTP Server does not supply default data set attributes. Instead, it gives SMS the ability to control the allocation of data sets. The FTP server only provides DCB parameters that the system administrator specified on the FTP DEFGAT statement, or that the user has explicitly specified with the SITE command.

For example, if SMS is specified, only allocation parameters from the FTP DEFGAT statements or the SITE command are used for dynamic allocation of new data sets. If SITE RESET is sent, then the values default either to your APPCFG*xx*-supplied values or to the specified defaults. See the following diagram for more details.

Flow of FTP when SMS Parameter Specified on HOST Statement

```
┌─────────────────────────┐   Yes   ┌─────────────────────────┐
│ Does the dataset exist? │────────▶│ Use existing DCB        │──────┐
└─────────────────────────┘         │ parameters              │      │
            │ No                     └─────────────────────────┘      │
            ▼                                                          │
┌─────────────────────────┐   No    ┌─────────────────────────┐      │
│ Has SMS been specified? │────────▶│ Supply default dataset  │─────▶│
└─────────────────────────┘         │ attributes              │      │
            │ Yes                    └─────────────────────────┘      │
            ▼                                                          │
┌─────────────────────────┐   Yes   ┌─────────────────────────┐      │
│ Have any dataset        │────────▶│ Use parameters from FTP │─────▶│
│ attributes been         │         └─────────────────────────┘      │
│ specified on the FTP    │                                           │
│ statement?              │                                           │
└─────────────────────────┘                                          │
            │ No                                                       │
            ▼                                                          │
┌─────────────────────────┐   Yes   ┌─────────────────────────┐      │
│ Has a DEFGAT been coded?│────────▶│ Use DCB parameters from │─────▶│
└─────────────────────────┘         │ DEFGAT                  │      │
            │ No                     └─────────────────────────┘      │
            ▼                                                          │
┌─────────────────────────┐   Yes   ┌─────────────────────────┐      │
│ Has the FTP user issued │────────▶│ Use dataset attributes  │─────▶│
│ an FTP SITE command?    │         │ from SITE command       │      │
└─────────────────────────┘         └─────────────────────────┘      │
            │ No                                                       │
            ▼                                                          │
┌─────────────────────────┐                                          │
│ Issue the dynamic       │◀─────────────────────────────────────────┘
│ allocation              │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│ Verify that SMS has set │
│ many/all of the DCB     │
│ parameters              │
└─────────────────────────┘
```

Buffers

The IBUF and OBUF values are negotiated by FTP with the API before the data port is opened. If the values exceed the API maximum values as specified in the TCP statement, they will be reduced. For instance, the actual buffer space that will be allocated for the receiving data from the network will be equal to the lower of the TCP MAXRCVBUF value, or to the following:

```
lower(FTP IBUF(numbuf), TCP MAXQRECV)
x lower(FTP IBUF(bufsize), TCP MAXLTRCV)
```

PDS Enqueue Parameters

Use the PDS enqueue parameters ISPFENQ, ISPFRES, and LKEDRES to comply with the OS/390 and ISPF conventions that ensure data set integrity when opening partitioned data sets in an UPDATE mode. With this feature, FTP provides for simultaneous access to a partitioned data set (PDS) during data transfer. That is, a user can browse or edit any members in the PDS while an FTP data transfer is in progress.

The PDS enqueue mechanism is activated whenever you execute an FTP STOR, APPEND, DELETE, or RENAME operation on members of PDS or PDS/E data sets. This mechanism allows concurrent FTP users to simultaneously access data stored as members of the same PDS.

Normal member locking mechanisms will be in place. If a member is open in edit mode and FTP attempts to access it, an error message is returned to the user with a notice that the member is in use.

The PDS enqueue mechanism follows the conventions described in IBM's document *Interactive System Productivity Facility (ISPF) Planning and Customizing*. This information is in Appendix B of Version 4 Release 1 for MVS.

FTP Examples

This example shows the usage of the FTP statement:

```
FTP  GUEST(DEMO1 NOPASSWD)
     ACCOUNT(ACCES 0000)
     PORT(21)
     UNIT(SYSDA)
     IBUF(10 2048)
     OBUF(10 2048)
     ACCTREQ
     NORECALL
     NDAB(8)
     OUTLIM(400000)
```

This example shows how to configure the FTP statement to support PDS Enqueue:

```
FTP  ISPFENQ
     ISPFRES
     LKEDRES
```

This example demonstrates the usage of additional parameters:

```
FTP   GUEST(GUEST XXXXXXXX)
      ACCOUNT(OVERHEAD 0000)
      PORT(21)
      IBUF(20 2048)
      OBUF(20 2048)
      LABEL(NOBLP,NONL)
      MOUNT(300,400)
      DEFGAT(FILE)
      AUTORECALL
      PAD(B B)
      UNIT(SYSALLDA)
      CHARSET(ENGLISH)
      TRANTBL(ENGLISH)
```

Generic Attributes Table (GAT)

The Generic Attributes Table (GAT) statements are a series of predefined attributes for certain types of file transfers. Use the GAT statement to specify data set attributes by data set type to be used by the FTP SITE command.

Caution! *The characteristics of this GAT can be changed, particularly the BLK and SPACE parameters. You can also create a new GAT statement and make it the default by pointing to it with DEFGAT.*

You must retain all the GAT statements distributed because Unicenter TCPaccess FTP Server expects them. New ones can be added and referenced in the FTP SITE parameter or the DEFGAT parameter of the FTP statement.

The FTP SITE command is documented in the *Unicenter TCPaccess FTP Server User Guide.*

```
GAT       TYPE ( name )
          [ BLKSIZE ( number ) ]
          [ COMPACT ]
          [ CONDDISP ( CATLG | DELETE ) ]
          [ DATACLAS ( data_class_name ) ]
          [ DCBDSN ( data_set_name ) ]
          [ EXPDT ( expiration_date ) | RETPD ( retention_period ) ]
          [ ISPFENQ | NOISPFENQ ]
          [ ISPFRES | NOISPFRES ]
          [ LABEL ( type ) ]
          [ LKEDRES | NOLKEDRES ]
          [ LRECL ( number ) ]
          [ MANAGEMENTCLAS ( management_class_name ) ]
          [ PARALLELMOUNT ]
          [ PDSE ]
          [ PRIVATE ]
          [ RECFM ( F | FA | FB | FBA | FBS | FBSA | V | VB | VM | VS | VBA |
                          VBS | VBSA | U ) ]
          [ SPACE ( CYL | TRK | BLK ( [ pri ] [ sec ] [ dir ] ) ) ]
          [ STORCLAS ( storage_class_name ) ]
          [ UNIT ( unitname ) ]
          [ UNITCOUNT ( count ) ]
          [ VOLUME ( volser ) ]
```

TYPE (*name*)   Specifies a data set type (one- to eight-characters) corresponding to the string that must be specified with an FTP SITE command. If TYPE(TAPE) is used, FTP MOUNT parameters must be set appropriately.

Default: None.

BLKSIZE (number)   Specifies the physical block size (0 to 65535) associated with the TYPE data set.

Default: 6080.

COMPACT   Specifies IDRC compaction for 3480 tapes.

CONDDISP(CATLG | DELETE)

Specifies the conditional disposition of new data sets when a STOR operation fails.

Default: CATLG.

DATACLAS (*data_class_name*)—Specifies the SMS data class.

DCBDSN (*data_set_name*)

Specifies the name of a data set to use as a model for DCB attributes when allocating new data sets.

Default: None.

EXPDT (*expiration_date*) | RETPD (*retention_period*)

Specifies expiration date or retention period for a new data set.

expiration_date   The format is *yyyyddd* or *yyyy/ddd*, where *yyyy* is a year from 1900 to 2155, and *ddd* is a Julian date from 1 to 366. You must include any leading zeroes in the *ddd* value.

retention_period Specifies a number of days between 1 and 9999.

Note: EXPDT and RETPD are mutually exclusive.

ISPFENQ | NOISPFENQ

Specifies that the ISPF enqueue facility be activated (ISPFENQ) or deactivated (NOISPFENQ). See the section, FTP Configuration Parameters - FTP Statement, for more parameters for using PDS enqueue.

Default: NOISPFENQ.

ISPFRES | NOISPFRES Enables (ISPFRES) or disables (NOISPFRES) the RESERVE logic for the SPFEDIT ENQ, if the volume on which the PDS resides is shared by Multiple Systems (UCB shared bit ON). This assures data integrity while the PDS you are accessing is being simultaneously accessed by an ISPF user from another system.

See the section, FTP Configuration Parameters - FTP Statement, for more parameters for using PDS enqueue.

Default: NOISPFRES.

LABEL (*type*) Specifies tape label type. These label options are supported:

SL   Standard labels.

NL   No labels.

BLP   Bypass label processing.

LTM   Leading tape mark.

AL   ASCII labels.

Default: SL.

LKEDRES | NOLKEDRES

Enables (LKEDRES) or disables (NOLKEDRES) the RESERVE logic for the SYSIEWLP ENQ, if the volume on which the PDS resides is shared by multiple systems (UCB shared bit ON). This assures data integrity while the PDS you are accessing is being simultaneously accessed by the linkage editor from another system.

See the section, FTP Configuration Parameters - FTP Statement, for more parameters for using PDS enqueue.

Default: NOLKEDRES.

LRECL (*number*) Specifies the logical record length (1 to 65535) associated with the TYPE data set.

Default: 80.

MANAGEMENTCLAS (*management_class_name*)

Specifies the SMS management class.

PARALLELMOUNT   Specifies that each volume of a data set be mounted on a separate device (mutually exclusive with UNITCOUNT).

Note: PARALLELMOUNT can be abbreviated to PARALLEL.

PDSE   Allocates PDSEs instead of PDSs.

PRIVATE   Requests private volume.

RECFM (F | FA | FB | FBA | FBS | FBSA | V | VB | VM | VS | VBA | VBS | VBSA | U)

Specifies the RECFM to be associated with the TYPE data set.

Default: FB.

SPACE (CYL | TRK | BLK ([*pri*] [*sec*] [*dir*]))

CYL | TRK | BLK—one of these must be coded as shown. It specifies allocation unit in cylinders, tracks, or blocks.

pri   Specifies the number of allocation units in the first allocation request for the TYPE data set.

sec   Specifies the number of allocation units for secondary requests for space, used when the primary request is exceeded.

dir   Specifies the number of 256-byte directory blocks to be reserved for a partitioned data set (PDS) directory. One block holds from 7 (load module) to 16 (source module) member entries. This parameter indicates a TYPE data set is a PDS.

Default: None.

STORCLAS (*storage_class_name*)

Specifies the SMS storage class.

UNIT (*unitname*)   Specifies a generic unit to be associated with the TYPE data set.

Default: None.

UNITCOUNT (*count*)   Specifies the number of devices to be allocated to the data set (mutually exclusive with PARALLELMOUNT). UNITCOUNT can be abbreviated to UCNT.

Default: One.

VOLUME (*volser*)   Specifies an explicit volume to be associated with the TYPE data set.

Default: None.

Usage Notes for the GAT Statement

You may need to change these parameters on the GAT statement:

TYPE(FILE)   This is the default GAT pointed to by the DEFGAT parameter in the FTP statement and it allows you to set the DCB and SPACE parameters.

PATH   The high-level qualifier for the PATH parameter must not be the same high-level qualifier that was used for the Unicenter TCPaccess FTP Server data sets.

RECFM, LRECL, and BLKSIZE DCB Parameters
Standard rules apply for the relationship between RECFM, LRECL, and BLKSIZE DCB parameters. These are validated in the GAT statement.

Necessary Types
GAT statements must be provided for these types: SOURCE, CARDS, FORTRAN, OBJECT, LOADLIB, TAPE, PRINT. Other statements and types can be added as required.

DEFGAT *Paramete*)
If a DEFGAT parameter is specified on the FTP statement, then a GAT statement with the corresponding TYPE must be coded.

TYPE(INTRDR)
A special GAT entry, TYPE(INTRDR), can be coded to implement an installation default for allocation parameters for the internal reader. When a SITE SUBMIT is received, FTP scans the GAT table for a TYPE(INTRDR) entry. This entry provides the file attributes for the internal reader. Only RECFM, LRECL, and BLKSIZE are taken. Other parameters are ignored.

If no TYPE(INTRDR) entry is found, then the *Unicenter TCPaccess FTP Server* defaults apply, that is, (RECFM=FB,LRECL=80,BLKSIZE=20000).

TYPE(LIBRARY)
A special GAT entry, TYPE(LIBRARY), can be coded to provide allocation defaults for the server FTP MKD (make directory) command.

SMS Control of Default Data
In order to have the default data under SMS control, the system administrator must configure the SMS parameter on the GLOBAL statement and must add the following to the GAT statement:

```
GAT   DATACLAS(xxx)            /* all default */
      STORCLAS(XXX)            /* allocations */
      MANAGEMENTCLAS(XXXX)     /* SMS parameters */
      PDSE                     /* all PDSs as PDSEs */
```

SPFENQ, ISPFRES, and
LKEDRES Parameters

The GAT parameters ISPFENQ, ISPFRES, and LKEDRES enable the PDS enqueue feature of *Unicenter TCPaccess FTP Server*. See the section, FTP Examples, for information and more parameters to use PDS enqueue.

GAT Examples

This example shows the usage of the GAT statement:

```
GAT TYPE(SOURCE)    RECFM(FB)    LRECL(80)    BLKSIZE(3120)
GAT TYPE(OBJECT)    RECFM(FB)    LRECL(80)    BLKSIZE(2960)
GAT TYPE(LOADLIB)   RECFM(U)     LRECL(0)     BLKSIZE(6144)
GAT TYPE(CARDS)     RECFM(FB)    LRECL(80)    BLKSIZE(3120)
GAT TYPE(PRINT)     RECFM(VBA)   LRECL(137)   BLKSIZE(19069)
GAT TYPE(ASM)       RECFM(FB)    LRECL(80)    BLKSIZE(3120)
GAT TYPE(CNTL)      RECFM(FB)    LRECL(80)    BLKSIZE(3120)
GAT TYPE(PL1)       RECFM(FB)    LRECL(80)    BLKSIZE(3120)
GAT TYPE(FORTRAN)   RECFM(FB)    LRECL(80)    BLKSIZE(3120)
GAT TYPE(SPECIAL)   RECFM(VBS)   LRECL(1000)  BLKSIZE(5004)
    UNIT(ADRIVE)    SPACE(BLK(100 200))
GAT TYPE(LOAD2)     RECFM(U)     LRECL(0)     BLKSIZE(6144)
    VOLUME(ABC123)  SPACE(TRK(31 45))
```

This example shows how to provide a facility for all data to be under SMS control:

```
GLOBAL SMS    /* Turn on SMS system wide */
GAT TYPE(SOURCE)
    RECFM(FB)
    DATACLAS(xxxx)
    STORCLAS(yyyy)
    MANAGEMENTCLAS(zzzz)
    PDSE
```

This example shows the setting for PDS enqueue.

```
GAT TYPE(FILE) ISPFENQ
GAT TYPE(FILE) ISPFRES
GAT TYPE(FILE) LKEDRES
```

FTP Support for SMF Activity Reporting and User Accounting

Unicenter TCPaccess FTP Server collects information about user access and protocol data activity, and records and formats this information for presentation to two different media. The first presentation is in the log that displays activity using messages and codes. The second presentation uses IBM's System Management Facility (SMF) data collection feature. Unicenter TCPaccess FTP Server uses user-provided parameters in IJTCFG*xx* to determine the degree, if any, of the SMF data recording desired. The SMF records provide useful information that can be displayed or analyzed by the SMF report writer program, T00SMFWR, or by programs provided by other suppliers.

The SMF Accounting Facility provides for the generation of SMF records by the FTP server component. The facility also allows the validation of account data that can be entered at any time using the ACCOUNT command or can be required at user logon through a parameter in T051CFA*x.*

An account exit routine is provided with the Unicenter TCPaccess FTP Server software. The exit routine can be used to refuse a logon request, and to call or load tables or other routines to assist in validating the logon request.

SMF record subtypes support the NLST, LIST, RETR, APPE, and STOR data transfer operations and RNFR/RNTO and DELE commands. User information, such as user identification and accounting information, is provided.

Defining SMF

To support generation of SMF records, use the SMF parameter in IJTCFG*xx* to set SMF recording on or off, the SMF record type, and the record subtypes desired. See the section, FTP Support for SMF Activity Reporting and User Accounting, for a description of this statement.

Defining FTP to Require Account Data

To specify that account data is required at user logon (in addition to password), add this ACCTREQ parameter on the FTP statement:

```
ACCTREQ | NOACCTREQ
ACCTREQ
```

You are prompted for account data at logon.

NOACCTREQ—Account data at logon is not required.

Running an SMP/E APPLY CHECK

You should run an SMP/E APPLY CHECK against any USERMOD that you are trying to install, as there may be additional PREs on your system that are not accounted for. After you gather this information, add the SYSMOD list(s) to the ++PRE(*xxxxxx*) statement. Then SMP/E REJECT the USERMOD to remove the invalid entry from the SMP/E CSI. You can then RECEIVE/APPLY the USERMOD with success.

Defining SMF Account Exit

Both source and object code are provided for the Unicenter TCPaccess FTP Server account exit, ACEXIT00. The source code is fully documented and annotated and located in the SAMP data set.

The exit program is loaded by Unicenter TCPaccess FTP Server at startup and resides in memory for the life of the Unicenter TCPaccess FTP Server job. It is called when a user has provided account information and the ACCTREQ parameter is coded on the FTP statement. The exit can use IBM assembler data management macro calls to load tables or other routines. The default exit is installed in the LINK data set and receives a 144-byte accounting field, SMFACACT, to record the accounting information.

The exit must have the CSECT name ACEXIT00. Use member UMODEX00 in the CNTL data set to install a new or modified exit.

The following is an example of UMODEX00.

```
//UMODEX00 JOB
//*
//*     SAMPLE JCL TO RECEIVE AND APPLY USERMOD TO REPLACE
//*     THE SUPPLIED ACCOUNTING EXIT ACEXIT00 WITH USER'S OWN.
//*
//*     GLOBALLY CHANGE THE FOLLOWING STRINGS TO REFLECT THE
//*     CORRECT SMPE DATASETS AND FMID.
//*
//*     'SMPINDX'    <SMP DATASETS HIGH LEVEL QUALIFIER
//*      XXX         <SMPE FMID IDENTIFING MVS TCP/IP FMID
//*
//*     VERIFY THAT THE JOB CARD AND NAMING CONVENTIONS MEET
//*     YOUR SITE'S JCL REQUIREMENTS, THEN SUBMIT THIS JOB.
//*
//SMPE     EXEC PGM=GIMSMP,REGION=4096K,TIME=960,
//         PARM='CSI=SMPINDX.CSI,PROCESS=WAIT'
//SMPHOLD  DD  DUMMY
//SMPLOG   DD  DSN=SMPINDX.SMPLOG,DISP=MOD
//SMPOUT   DD  SYSOUT=HOLDCL
//SMPPTFIN DD  *
++ USERMOD (MU0EX00) .
++ VER (Z038)
   FMID(T0XXXX)  /* CHANGE TO CORRECT TCP/IP FMID */ .
++ SRC  (ACEXIT00)  TXLIB(TCPSAMP) DISTMOD(ATCPLOAD)
DISTLIB(ATCPSAMP) .
```

```
/*
//SMPCNTL  DD  *
  SET BDY(GLOBAL) .
  RECEIVE S(MU0EX00) .
  SET BDY(TCPTZN) .
  APPLY   S(MU0EX00) .
/*
```

FTPRSRCE and FTPLOGIN Exits

Unicenter TCPaccess FTP Server allows you to define global user exits. The
FTPRSRCE exit point is used when an FTP command has been received which
will cause a data set allocation. The FTPLOGIN exit is defined for when an FTP
login is received.

Client FTP Configuration Data Sets

If you use Client FTP with a port other than port 21 (the default), then you need
to change the PORT parameter in the FTP.DATA configuration data set.

For further information about this data set, see the section, Understanding the
Configuration Data Sets, in the *User Guide*.

Chapter 4

Using the System Management Facility (SMF)

This chapter provides samples of SMF record formats and contains the following topics:

- Standard OS/390 SMF Record Header
- Standard Unicenter TCPaccess FTP Server SMF Record Header
- Server Application Event Records

Unicenter TCPaccess FTP Server generates SMF records in accordance with the record formats provided in IBM's *GC28-1030* manual. The Unicenter TCPaccess FTP Server SMF record is variable length but begins with the OS/390 SMF record header and the Unicenter TCPaccess FTP Server SMF record header. The Unicenter TCPaccess FTP Server SMF record header is the anchor block used to determine which other sections are present.

On the Unicenter TCPaccess FTP Server software tape, there is a sample of the JCL to dump Unicenter TCPaccess FTP Server SMF records from dump data sets. In the MAC library, the SMFRECS macro maps the SMF records.

Unicenter TCPaccess FTP Server SMF recording is specified by the SMF statement in member T051CFG*x* described in the chapter "Configuring FTP."

Standard OS/390 SMF Record Header

The following table provides a sample of the standard OS/390 record header.

Offset		Field	Length	Format	Source	Description
0	0	SMFACLEN	2	Binary	Internal	Record length
2	2	SMFACSEG	2	Binary	Internal	Segment descriptor
4	4	SMFACFLG	1	Binary	SVC 83	Header flag byte
5	5	SMFACRTY	1	Binary	Internal	Record type
6	6	SMFACTME	4	Binary	SVC 83	Time, in hundredths of a second, that the record was moved to the SMF buffer
10	A	SMFACDTE	4	Packed	SVC 83	Date record was moved to SMF buffer, in the form 00YYDDF, where F is the sign byte
14	E	SMFACSID	4	EBCDIC	SMCASID	System ID (taken from the SID parameter)

Standard Unicenter TCPaccess FTP Server SMF Record Header

This section provides samples of standard Unicenter TCPaccess FTP Server SMF record headers. The record headers include Descriptor and Product sections.

Descriptor

The following table shows the descriptor section of the FTP server record header.

Offset		Field	Length	Format	Source	Description
18	12	SMFACNOD	2	Binary	Internal	Number of descriptors in this section
20	14	SMFACPOF	4	Binary	Internal	Offset to product section
24	18	SMFACPLN	2	Binary	Internal	Length of product section
26	1A	SMFACPNO	2	Binary	Internal	Number of product sections
28	1C	SMFACTOF	4	Binary	Internal	Offset to task information section
32	20	SMFACTLN	2	Binary	Internal	Length of task information section
34	22	SMFACTNO	2	Binary	Internal	Number of task information sections
36	24	SMFACUOF	4	Binary	Internal	Offset to user identification section
40	28	SMFACULN	2	Binary	Internal	Length of user identification section
42	2A	SMFACUNO	2	Binary	Internal	Number of user identification sections
44	2C	SMFACDOF	4	Binary	Internal	Offset to data section
48	30	SMFACDLN	2	Binary	Internal	Length of data section
50	32	SMFACDNO	2	Binary	Internal	Number of data sections

Product

The following table shows the product section of the FTP server record header.

Offset		Field	Length	Format	Source	Description
0	0	SMFACTYP	2	Binary	Internal	Subtype identification
2	2	SMFACRVN	2	EBCDIC	Internal	Record version number – 02
4	4	SMFACNAM	8	EBCDIC	Internal	Product name
12	C	SMFACVER	2	EBCDIC	Internal	Product version number
14	E		8			Reserved
22	16	SMFACJBN	8	EBCDIC	Internal	Jobname of Unicenter TCPaccess FTP Server job
30	1E	SMFACJBI	8	EBCDIC	Internal	JES job ID of Unicenter TCPaccess FTP Server job
38	26	SMFACASI	2	Binary	Internal	Address space ID of Unicenter TCPaccess FTP Server job
40	28	SMFACSSI	4	EBCDIC	Internal	Subsystem ID of Unicenter TCPaccess FTP Server job

Server Application Event Records

Task Identification Section

The following table contains the task identification section (pseudo-task data) of the Unicenter TCPaccess FTP Server SMF header.

Offset		Field	Length	Format	Source	Description
0	0	SMFACPGM	8	EBCDIC	PTAPGM	Program name

User Identification Section

The following table contains the user identification section (AC#U data) of the Unicenter TCPaccess FTP Server SMF header.

Offset		Field	Length	Format	Source	Description
0	0	SMFACUID	8	EBCDIC	SECUID	RACF (or other external security system) user ID
8	8	SMFACGID	8	EBCDIC	SECGRP	RACF (or other external security system) group ID
16	10	SMFACACT	144		AC#UACCT	Accounting information

The following is the format of the SMFACACT field:

■   First byte of field contains the number (binary) of accounting fields

■   Following bytes contain accounting fields, where each entry for an accounting field contains a one-byte length field, followed by the field

FTP Server Records (Subtypes 20 - 22)

Descriptor

The following table shows the descriptor section of the FTP server record header.

Offset		Field	Length	Format	Source	Description
0	0		2	Binary	Internal	Reserved
2	2	FTPDNOD	2	Binary	Internal	Number of descriptors in this section
4	4	FTPDCSOF	4	Binary	Internal	Offset to FTP server common data section
8	8	FTPDCSLN	2	Binary	Internal	Length of FTP server common data section
10	A	FTPDCSNO	2	Binary	Internal	Number of FTP server common data sections
12	C	FTPDDTOF	4	Binary	Internal	Offset to FTP server data transfer completion data section
16	10	FTPDDTLN	2	Binary	Internal	Length of FTP server data transfer completion data sections
18	12	FTPDDTNO	2	Binary	Internal	Number of FTP server data transfer completion data sections

Common Data

The following table shows the common data section of the FTP server record header.

Offset		Field	Length	Format	Source	Description
0	0	FTPCRHST	4	Binary	CACCRHST	Remote host Internet address for control connection
4	4	FTPCLHST	4	Binary	CACCLHST	Local host Internet address for control connection
8	8	FTPCRPRT	2	Binary	CACCRSCK	Remote host TCP port number for control connection
10	A	FTPCLPRT	2	Binary	CACCLSCK	Local host TCP port number for control connection

FTP Data Transfer Completion/End of Volume (Subtype 20/Subtype 22) Data Section

The Subtype 20 record is written by the FTP server at the end of a data transfer.

The Subtype 22 record is written at end-of-volume during a data transfer of a multi-volume data set. Its layout is identical to that of the Subtype 20 record.

The following table provides a sample of the Subtype 20/Subtype 22 data section.

Offset		Field	Length	Format	Source	Description
0	0	F20DTTY	1	Binary	CAFLGA1	Type of data transfer operation: 1 = NLST (data set name list without details) 2 = LIST (data set name list with details) 8 = RETR (retrieve) 16 = APPE (append) 32 = STOR (store)
1	1		1	Binary	Internal	Reserved
2	2	F20TYPE	1	Binary	CATYPE	Data type (from FTP TYPE command): 0 = ASCII 4 = EBCDIC 8 = Binary 16 = local byte
3	3	F20FORM	1	Binary	CAFORM	Format control (from FTP TYPE A or E command): 0 = None 4 = Telnet 8 = ASA CC
4	4	F20STRUA	1	Binary	CASTRU	Data structure (from FTP STRU command): 0 = File 4 = Record

Offset		Field	Length	Format	Source	Description
5	5	F20MODE	1	Binary	CAMODE	Transmission mode (from FTP MODE command): 0 = Stream 4 = Block 8 = Compressed
6	6	F20DSN	44	EBCDIC	CAPADSN	Data set name: RETR APPE STOR Or path name: NLST LIST
50	32	F20MEM	8	EBCDIC	CAPAMEM	Member name
58	3A	F20DSSN	6	EBCDIC	CAPADSSN	Data set serial number (the first or only *volser* for the data set)
58	3A	F20VOL	6	EBCDIC	CAPADSSN	Same as F20DSSN. (This field kept for compatibility)
64	40	F20DDNM	8	EBCDIC	CADDNAME	DD name
72	48	F20RHST	4	Binary	CADTRHST	Remote host Internet address for data connection
76	4C	F20LHSTA	4	Binary	CADTLHST	Local host Internet address for data connection
80	50	F20RPRT	2	Binary	CADTRSCK	Remote host TCP port number for data connection
82	52	F20LPRT	2	Binary	CADTLSCK	Local host TCP port number for data connection
84	54	F20STRT	4	Binary	CATIME	Time, in hundredths of a second, that data transfer started
88	58	F20DURA	4	Binary	CATIME+4	Elapsed time, in hundredths of a second, for data transfer operation
92	5C	F20MSG	2	Binary	CAREPFTP	Last FTP reply number
94	5E	F20SBMSG	2	Binary	CAREPSUB	Sub-reply number for last FTP reply

Offset		Field	Length	Format	Source	Description
96	60	F20DAIR	4	Binary	CADAIR	PDYNAL reason code:
						SVC 99 reason code if RETR, APPE, or STOR
						SUPERLOCATE reason code if NLST or LIST
100	64	F20DARC	1	Binary	CADARC	PDYNAL return code:
						SVC 99 return code if RETR, APPE, or STOR
						SUPERLOCATE return code if NLST or LIST
101	65		3	Binary	Internal	Reserved
104	68	F20DSKB	4	Binary	CABYTES	Number of disk bytes read or written
108	6C	F20NETB	4	Binary	CALBYTES	Number of network bytes sent or received
112	70	F20RECT	4	Binary	CATRUNCS	Number of records truncated
116	74	F20RECP	4	Binary	CAPADDS	Number of records padded
120	78	F20RECF	4	Binary	CAFOLDS	Number of records folded
124	7C	F20RECS	4	Binary	CASUSP	Number of records suspected
128	80	F20RSTM	4	Binary	CARSTMKS	Number of restart markers sent or received
132	84	F20RSTI	4	Binary	CARSTIGN	Number of restart markers ignored
136	88	F20AERR	4	Binary	CA#TERR	API composite return code
140	8C	F20EXPD	7	EBCDIC	CAPAEXPD	Expiration date
147	93	F20RETP	2	Binary	CAPARETP	Retention period
149	95	F20VOLI	6	EBCDIC	CACRVOL	Volume serial number. For subtype 22 records, this is the volume at EOV. For subtype 20 records, this is the last volume for the data set.

Offset		Field	Length	Format	Source	Description
155	9B	F20VLSQ	2	Binary	CACRVLSQ	Volume sequence number.
						For subtype 22 records, this is the sequence number of the volume at EOV.
						For subtype 20 records this is the number of volumes for the data set.
155	9B	F20NVOL	2	Binary	CACRVLSQ	Same as F20VLSQ
157	9D	F20DACL	30	Character	CAPADACL	SMS Data class
187	BB	F20STCL	30	Character	CAPASTCL	SMS Storage class
217	D9	F20MGCL	30	Character	CAPAMGCL	SMS Management class

Note: The subtype 20 and subtype 22 records have identical formats.

The totals fields in the subtype 22 EOV records (for example, F20DURA—elapsed time—and F20DSKB—disk bytes written) are calculated from the beginning of the data transfer (not the beginning of processing for the volume).

FTP Data Set Modification (Subtype 21) Data Section

The subtype 21 record is written by the FTP server when a request to modify a data set (other than a data transfer) is completed. This includes the rename, delete, and make directory (MKD) commands.

The following table provides a sample of the Subtype 21 data section.

Offset		Field	Length	Format	Source	Description
0	0	F21DMTY	1	Binary	CAFLGS1	Data set modification type 16 = MKD (make directory) 64 = DELE (delete) 128 = RENM/RNTO (rename)
1	1		1	Binary	Internal	Reserved
2	2	F21DDNM	8	EBCDIC	CADDNAME	DD name
10	A	F21FDSN	44	EBCDIC	CAPADSN	Rename from or delete data set name
54	36	F21FMEM	8	EBCDIC	CAPAMEM	Rename from or delete member name
62	3E	F21DSSN	6	EBCDIC	CAPADSSN	Data set serial number (the first or only *volser* for the data set)
62	3E	F21VOL	6	EBCDIC	CAPADSSN	Same as F21DSSN (this field kept for compatibility)
68	44	F21TDSN	44	EBCDIC	CAPADSN	Rename to data set name
112	70	F21TMEM	8	EBCDIC	CAPAMEM	Rename to member name
120	78	F21MSG	2	Binary	CAREPFTP	Last FTP reply number
122	7A	F21SBMSG	2	Binary	CAREPSUB	Sub-reply number for last FTP reply
124	7C	F21DAIR	4	Binary	CADAIR	PDYNAL reason code
128	80	F21DARC	1	Binary	CADARC	PDYNAL return code
129	81	F21NVOL	2	Binary	CAPANVOL	Number of volumes
131	83	F21VOLS	8-2040	Structure		Beginning of the volume section (This section is not present for data sets residing on a single volume)

For each volume of a multi-volume data set, there is an eight-byte section with the format that is shown in the following table.

Offset		Field	Length	Format	Source	Description
0	0		2	Binary		Reserved
2	2	F21VOLI	6	EBCDIC	CAPAVOLS	Volume serial number

User Exits

This chapter provides information about writing exit routines for Unicenter TCPaccess FTP Server.

The Unicenter TCPaccess FTP Server exit facility lets users write exit routines to handle certain specialized requirements within their installation.

Various user exit points are defined within Unicenter TCPaccess FTP Server to allow the product to be customized.

Multiple exit programs can be configured for the various exit points. An exit program can communicate with itself across various exit point invocations by establishing an exit context. Exit programs can issue messages, accept or reject various requests, and change or reroute messages. Exit points are defined in message services and in FTP.

The user exit points listed in the following table are defined. These exits are configured in the T051CFI*x* member of the PARM dataset. For details about configuring the exit points, see the section, Setting Exit Points.

Exit Point	When Invoked	Function
INIT	Startup	Initialize the exit environment.
TERM	Shutdown	Terminate the exit environment.
LOG	A message is formatted.	The exit can change the message text, reroute the message, or suppress the message.
FTPCMND	An FTP command is received.	The exit can reject the command.
FTPLOGIN	An FTP login is received.	The exit can reject the request.
FTPRSRCE	An FTP command has been received which will cause a dataset allocation.	The exit can reject the request.

Parameters

Except where noted in the following discussion, parameter lists and the data areas they point to should be left unchanged by the exit program. Changes to other fields will be ignored and will not be made effective.

Exit Point ID

Each exit is passed a parameter list pointed to by R1, which includes a full word identifying the exit point. The first word of the parameter list is one of the following:

F'0'	INIT exit
F'1'	TERM exit
F'2'	LOG exit
F'14'	FTPLOGIN exit
F'15'	FTPRSRCE exit
F'22'	SMF exit

Issuing Messages from Exits

Each exit except the LOG exit will be passed the address of a routine that can be invoked in order to write a message to the log or to the operator. When calling this routine, the following must be provided:

R00	Must contain the value of R13 on entry to the exit program.
R01	Must point to a one-byte message type, followed by eighty bytes of message text. For a description of the message types, see the *Unicenter TCPaccess Prefixed Messages* manual and the *Unicenter TCPaccess Unprefixed Messages and Code*s manual.
R13	Must point to a standard 72-byte register save area.
R14	Must contain the return address.

The exit facility prefixes the messages with a standard Unicenter TCPaccess FTP Server message ID (T00EX004) and the exit program name.

Exit Context

At each exit point except INIT, the program is passed a full word of context. This context word is provided by the program at the INIT exit point, and can be used by the exit program to communicate across exit points.

Return Codes

There are two return codes defined for exits. The first, Return Code 1, is returned by the exit in R15 when returning to the exit facility. This return code determines whether the next configured exit program will be called for this exit point, if multiple exits are configured. This return code should normally be set to zero to allow subsequent exit programs to be invoked. A non-zero value will cause the exit facility to bypass calling any subsequent exit programs configured at that exit point. (Return code 1 does not apply to the INIT and TERM exits. Register 15 should be set to zero when returning from these exit points.)

The second return code, Return Code 2, is part of the parameter list provided on entry to the exit. The exit can set this return code, when appropriate, to reject the current request. For instance, the FTPRSRCE exit can set a non-zero value in this word to reject the FTP request. If multiple exit programs are configured at an exit point, the Return Code 2 value is passed from one exit program to the next. The value in Return Code 2 after the last exit is called is the value that will be returned to the caller of the exit facility. An exit program, therefore, should not change the Return Code 2 value, and, in particular, should not change it from a non-zero to a zero value, unless there is a compelling reason.

Care should be used when setting return codes and when configuring exits. The exit programs will be invoked in the same order in which they are configured. A subsequent program can change the Return Code 2 setting from an earlier exit program. An earlier exit program can prevent a subsequent exit program from being called by using Return Code 1.

The Exits

This section describes the individual exits.

There is a sample EXIT program in the *hlq*.SAMP data set.

INIT Exit

Exit Point: Unicenter TCPaccess FTP Server startup.

Function: The exit is called synchronously at startup, during parsing of the configuration statements in the T051CFC0 configuration member. It can be used to create an exit context which will be passed to the exit program at subsequent exit points. It can also issue messages.

Addressing Mode: 31

Dispatchable Unit: Task mode (TCB)

Restriction: The exit should not block indefinitely.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Zero

INIT Exit Parameters Passed

Offset	Parm Length	Data Length	Description
+00	04	—	Exit point ID. This word will contain F'0'.
+04	04	—	Exit context. This word will contain zeros on input. The INIT exit can place a word of context in this word. The context will then be passed to the exit program at other exit points.
+08	04	—	Return code 2. This should be set to zero.
+12	04	—	Address of the message routine. Note, since the log is not allocated when this exit is called, the message type should be one that will be written by WTO. The LOG exit point can be used to ensure that the message written (T01EX004) is routed to the console.
+16	04	Var	Address of a word in which the exit may define which exit points will be driven (except the TERM exit, which is always driven). This word can be built by ORing the exit point ID flag values for the exits which are to be driven. Exit point flags are defined as follows: l X'80000000' - LOG exit l X'00800000' - TCPBIND exit l X'00000080' - FTPLOGIN exit l X'00000040' - FTPRSRCE exit For instance, to drive the FTPLOGIN and FTPRSRCE exits, set the value of this word to X'000000C0'. Note that this request may be overridden by the configuration. See the discussion of the EXIT statement in.
+20	04	04	Address of the PARM string from the EXIT configuration statement. Note that this area will be released following the exit point. If the exit program wishes to save this string, it must obtain storage and make a copy.
+24	0	Var	Address of a full word containing the length of the PARM string from the EXIT configuration statement.

TERM Exit

Exit Point: Unicenter TCPaccess FTP Server shutdown.

Function: The exit is called synchronously at shutdown. It can be used to terminate the exit environment and clean up.

Addressing Mode: 31

Dispatchable Unit: Task mode (TCB)

Restrictions: The exit should not block indefinitely.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Zero

TERM Exit Parameters Passed

Offset	Parm Length	Data Length	Description
+00	04	—	Exit point ID. This word will contain F'1'.
+04	04	—	Exit context.
+08	04	—	Return code 2. This should be left unchanged.
+12	04	—	Address of the message routine.

LOG Exit

Exit Point: When a message has been formatted and is ready to be written.

Function: The exit can change the message text, reroute the message, or suppress the message.

Addressing Mode: 31

Dispatchable Unit: Task mode PC routine or SRB

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return code 1

LOG Exit Parameters Passed

Offset	Parm Length	Data Length	Description
+00	04	—	Exit point ID. This word will contain F'2'.
+04	04	—	Exit context.
+08	04	—	Return code 2. This should be left unchanged. To suppress the message, set the message routing code to zero (see below).
+12	04	—	Zeros
+16	04	Var	Address of the message text. The text may be changed by the exit.
+20	04	04	Address of a word containing the message buffer length. (This value should not be changed; the message cannot be made any longer than this length. To shorten the message, pad to the right with blanks.)
+24	0	Var	Address of a word containing a message routing code: X'00000000' = Do not issue message X'00000004' = Write message to log X'00000008' = Write message to the operator console (WTO) X'0000000C' = Write to log and console. This word may be changed by the exit to change the routing for the message. To suppress the message, set this word to zero.

FTP Exits

Following are the FTP exit points. These exits are invoked in task mode, but *must not* block execution on the TCB.

These exits can be used to issue a message or reject an access request.

FTPLOGIN Exit

Exit Point: When an FTP user is logging in (when the PASS or ACCT command is received). The exit is called before the SAF router is called to log the user on.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode.

Restrictions: The exit must not block execution.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return code 1

FTPLOGIN Exit Parameters Passed

Offset	Parm Length	Data Length	Description
+00	04	—	Exit point ID. This word will contain F'14'.
+04	04	—	Exit context.
+08	04	—	Return code 2. The exit can set this word to a non-zero value to reject the login.
+12	04	—	Address of the message routine.
+16	04	08	Address of the local host AF_Inet.
+20	04	08	Address of the remote host AF_Inet.
+24	04	08	Address of the user ID.

FTPRSRCE Exit

Exit Point: When an FTP user has requested an operation which will result in a data set resource allocation. This exit is called before the SAF router is invoked to verify the user's access authority.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode.

Restrictions: The exit must not block execution.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return code 1

FTPRSRCE Exit Parameters Passed

Offset	Parm Length	Data Length	Description
+00	04	–	Exit point ID. This word will contain F'15'.
+04	04	–	Exit context.
+08	04	–	Return code 2. The exit can set this word to a non-zero value to reject the request.
+12	04	–	Address of the message routine.
+16	04	08	Address of the local host AF_Inet.
+20	04	08	Address of the remote host AF_Inet.
+24	04	52	Address of the 44-byte dataset name, followed by an eight-byte member name (or blanks).
+28	04	06	Address of the volume serial number, or blanks.
+32	04	01	Address of a one-byte field identifying the authorization request type: X'40' - Read X'20' - Write X'10' - Alter
+36	04	02	Address of a two-byte field identifying the FTP operation type: X'8000' - HELP command X'4000' - STAT command (Statistics) X'2000' - STOR command (Store) X'1000' - APPE command (Append) X'0800' - RETR command (Retrieve) X'2400' - STOU command (Store Unique) X'0200' - LIST command (Directory list) X'0100' - NLST command (Directory list)

Offset	Parm Length	Data Length	Description
			X'0080' - RNTO command (Rename)
			X'0040' - DELE command (Delete)
			X'0020' - Special PDS directory allocation for LIST/NLST
			X'0010' - MKD command (Make Directory)
+40	04	Var	Address of the ACEE. The ACEE is mapped by macro IHAACEE. Its length is in the ACEELEN field.

Chapter

6

Troubleshooting the FTP Client and Server

This chapter describes problem isolation in the FTP client and server. Troubleshooting information is divided into server and client. The client is a three-party model; it therefore requires both server and client documentation, it contains the following topics:

- General Guidelines for Problem Isolation

- Resolving Server FTP Problems

- Resolving Client FTP Problems

General Guidelines for Problem Isolation

When encountering FTP3 problems, start with the following problem resolution process.

1. Determine the level of maintenance that is running.

2. Enable the DEBUG or TRACE option and rerun the failing command.

3. Run a network trace.

4. Look at the T01LOG job output.

Resolving Server FTP Problems

For incorrect input through FTP, collect the following:

- A description of the action preceding the failure and the condition of the system (such as 100-terabyte transfer)

- The job output of the Unicenter TCPaccess FTP Server gateway at the time of failure, including
  - The JES log
  - Any PSNAP data
  - T01LOG
  - Any enabled traces

- A log or screen print of the failing transfer, including any messages

- Any client software involved, including the name of the vendor, the release number, the direction of transfer, the file format; and the contents (if requested)

- Whether the failure is reproducible or random. If it is reproducible, and the local stack is TCPaccess, then obtain a TCPEEP TCP trace.

Resolving Client FTP Problems

For incorrect input through FTP, collect the following information:

- A description of the action preceding the failure and the condition of the system (such as 100-terabyte transfer)

- The job output of the Unicenter TCPaccess FTP Server gateway at the time of failure, including
  - The JES log
  - Any PSNAP data
  - T01LOG
  - Any enabled traces

- A log or screen print of the failing transfer, including any messages

- Turn on the TRACE or DEBUG parameters. The recommended procedure is to run the command in batch mode and save the recorded trace output for problem resolution.

- Any client software involved, including the name of the vendor, the release number, the direction of transfer, the file format, and the contents (if requested)

- Whether the failure is reproducible or random. If it is reproducible, and the local stack is TCPaccess, then obtain a TCPEEP TCP trace.

Index

TYPE, 3-35, 3-38
UNIT, 3-37
UNITCOUNT, 3-37
VOLUME, 3-38
GLOBAL statement
MIGVOL, 3-14
SMS, 3-14
SVC99WTO, 3-14
TRANTBL, 3-15
IFSPARM statement
APPLICATIONS, 3-2
GTFID, 3-2
PROMPT, 3-3
TIMER, 3-3
TIMEZONE, 3-3
TRACENAME, 3-3
VSREPORT, 3-3
SERVICE statement
MODULE, 3-16
NAME, 3-16
PORT, 3-16
QLISTEN, 3-16
SMF statement
SUBTYPE, 3-11
TYPE, 3-11

password validation, 2-1

PATH parameter, 3-38

PDS enqueue, 3-33, 3-39

PDSE parameter, 3-37

persistent SITE, 3-26

port number configuration, 3-16

PORT parameter
FTP statement, 3-27
SERVICE statement, 3-16

PRIVATE parameter, 3-37

PROMPT parameter, 3-3

Q

QLISTEN parameter, 3-16

queued listen table, 3-16

R

RACF security, 2-8

RCVRID parameter, 3-27

RDW parameter, 3-27

RECFM parameter, 3-37

RELEASE parameter, 3-27

retention period, 3-35

RETPD parameter, 3-35

S

SAF, 2-1
processing, 2-3
security, 2-5

SAFPROT records, 2-4

security
data set RACF, 2-8
eTrust CA-ACF2
Version 5.2 or earlier, 2-3
Version 6 or later, 2-4
eTrust CA-Top Secret, 2-6
RACF, 2-8
SAF interface, 2-1
signon
CA-Top Secret, 2-6
eTrust CA-ACF2, 2-2
RACF, 2-8
source level, 2-6
eTrust CA-ACF2, 2-2
eTrust CA-Top Secret, 2-6
RACF, 2-8

server FTP
JES support
JESFILTER, 3-23
JESPUTGETTO, 3-23
troubleshooting, 6-1

SERVICE statement, 3-16

services, user-level, 3-16

signon security
eTrust CA-Top Secret, 2-6
eTrust CA-ACF2, 2-2

single byte character set, 3-19